MIPI DSI/CSI-2 to Parallel Bridge

Reference Design

FPGA-RD-02213-1.2

September 2025

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 all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. 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. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. 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

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviations	Definition
AXI4	Advanced eXtensible Interface 4
BPC	Bits Per Color
CRC	Cyclic Redundancy Check
CSI-2	Camera Serial Interface 2
D-PHY	Display PHY (Physical Layer)
DE	Data Enable
DPI	Display Pixel Interface
DSI	Display Serial Interface
EBR	Embedded Block RAM
EoTp	End of Transmission Packet
FIFO	First In, First Out
FPGA	Field Programmable Gate Array
GPLL	General Purpose PLL
HDL	Hardware Description Language
HS	High Speed
HSYNC	Horizontal Sync
IP	Intellectual Property
LP	Low Power
LUT	Look Up Table
LVCMOS	Low Voltage CMOS
LVDS	Low Voltage Differential Signaling
MIPI	Mobile Industry Processor Interface
MSB	Most Significant Bit
PLL	Phase Locked Loop
RAW	Raw Sensor Data
RGB	Red Green Blue
RX	Receiver
STA	Static Timing Analysis
TX	Transmitter
VSYNC	Vertical Sync

1. Introduction

The Mobile Industry Processor Interface (MIPI®) D-PHY is developed primarily to support camera and display interconnections in mobile devices, and has become the industry primary high-speed PHY solution for these applications in smartphones. MIPI D-PHY is typically used in conjunction with MIPI camera serial interface-2 (CSI-2) and MIPI display serial interface (DSI) protocol specifications. It meets the demanding requirements of low power, low noise generation, and high noise immunity for mobile phone designs.

MIPI D-PHY is a practical PHY for typical camera and display applications. It is designed to replace traditional parallel bus based on LVCMOS or LVDS. However, many processors and displays/cameras still use RGB, CMOS, or MIPI display pixel interface (DPI) as interface.

The MIPI-to-Parallel reference design allows the quick interface for a processor with a MIPI DSI interface to a display with an RGB interface or a camera with a MIPI CSI-2 interface to a processor with parallel interface. The Lattice Semiconductor MIPI-to-Parallel reference design provides this conversion for Lattice Semiconductor CrossLink™-NX devices. This is useful for wearable, tablet, human machine interfacing, medical equipment, and many other applications.

1.1. Quick Facts

Download the reference design files from the MIPI DSI/CSI-2 to Parallel Bridge Reference Design web page.

Table 1.1. Summary of the Reference Design

General	Target Devices	LIFCL-40
	Source Code Format	Verilog, System Verilog
Simulation	Functional Simulation	Performed
	Timing Simulation	Performed
	Testbench	Available
	Testbench Format	Verilog
Software Requirements	Software Tool and Version	Lattice Radiant™ software 2025.1
	IP Version	Byte-to-Pixel Converter IP v1.9.1 CSI-2/DSI D-PHY Rx IP v2.0.0

1.2. Features List

The key features of the MIPI-to-Parallel reference design are as follows:

- Compliant with MIPI D-PHY v1.2, MIPI DSI v1.2, and MIPI CSI-2 v1.2 specifications
- Supports MIPI D-PHY interfacing from 80 Mbps up to 2.5 Gbps
- Supports 1, 2, or 4 data lanes and one clock lane
- Supports continuous and non-continuous MIPI D-PHY clock
- Supports common MIPI DSI compatible video formats (RGB888, RGB666)
- Supports common MIPI CSI-2 compatible video formats (RGB888, RAW8, RAW10, RAW12)
- Supports MIPI DSI Video Mode operation of Burst and Non-Burst Events modes
- Supports dedicated End of Transmission short packet (EoTp)

1.3. Naming Conventions

1.3.1. Nomenclature

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

1.3.2. Signal Names

- *_n* are active low signals (asserted when value is logic 0)
- *_i* are input signals
- *_o* are output signals

1.3.3. Data Ordering and Data Types

The highest bit within a data bus is the most significant bit.

1-bit data stream from each MIPI D-PHY data lane is deserialized into 8-bit parallel data where bit 0 is the first received bit.

Table 1.2 lists the pixel data order from the core module.

Table 1.2. Pixel Data Order

Data Type	Format
RGB	{Red[MSB:0], Green[MSB:0], Blue[MSB:0]}
RAW	RAW[MSB:0]

2. Directory Structure and Files

Figure 2.1 shows the directory structure.

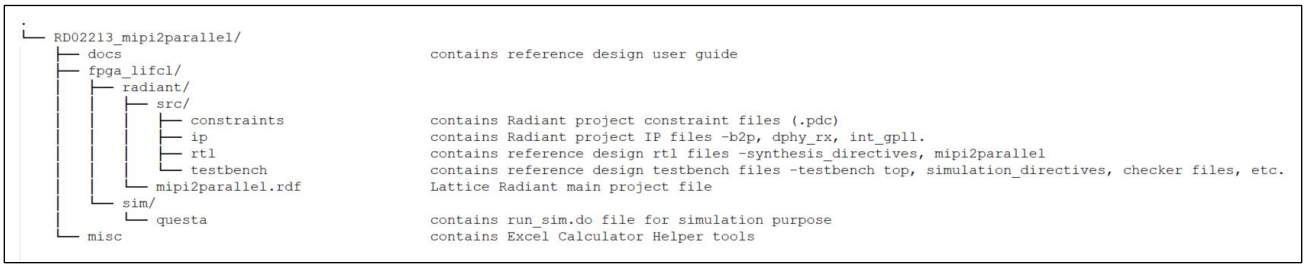

Figure 2.1. Directory Structure

3. Functional Description

Figure 3.1 shows the block diagram of the MIPI-to-Parallel reference design.

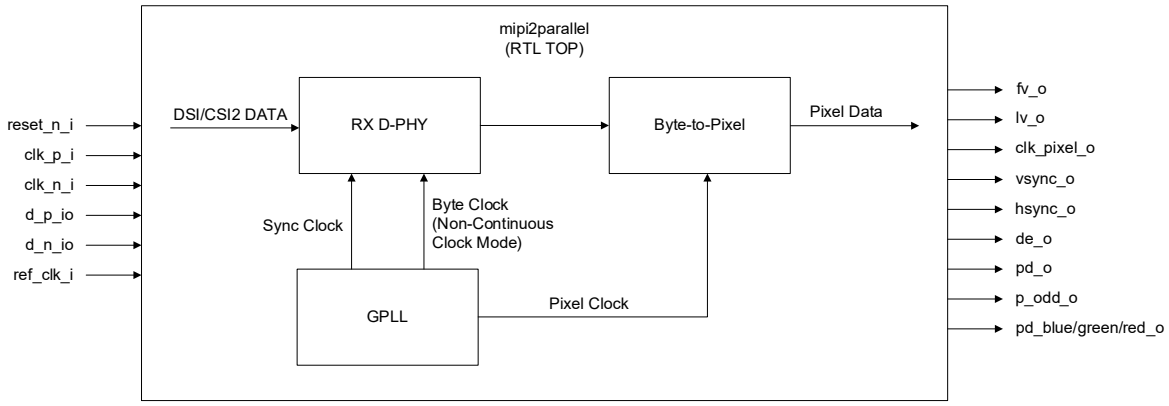

Figure 3.1. MIPI-to-Parallel Reference Design Block Diagram

As shown in Figure 3.1, the block diagram of the MIPI-to-Parallel reference design mainly consists of the RX D-PHY and Byte-to-Pixel IPs. There are two primary clocks for the main video data path: byte clock and pixel clock. Generally, a GPLL is used to generate the pixel clock. The same GPLL can also be used to generate the continuous byte clock when the RX D-PHY is in non-continuous clock mode. The input to the GPLL is sourced externally via the on-board oscillator or can be sourced internally with an oscillator, where you must instantiate an oscillator in the design. Sync clock (> 60 MHz) is required for the RX Soft D-PHY IP and a clock is required to drive LP (Low Power) HS (High Speed) mode detection logic in non-continuous clock mode.

The MIPI D-PHY receive interface has one clock lane and configurable number of data lanes. The clock lane is centre-aligned to the data lanes. The MIPI D-PHY clock can either be continuous (high speed only) or non-continuous.

When the MIPI D-PHY clock is non-continuous, proper transition from low power (LP) to high speed (HS) mode of clock lane is required. The data lanes also require proper transition from LP to HS modes. In HS mode, data stream from each data lane is deserialized to byte data. The deserialization is done with 1:8 gearing. The byte data is word-aligned based on the SoT Sync sequence defined in the *MIPI D-PHY Specification version 1.2*.

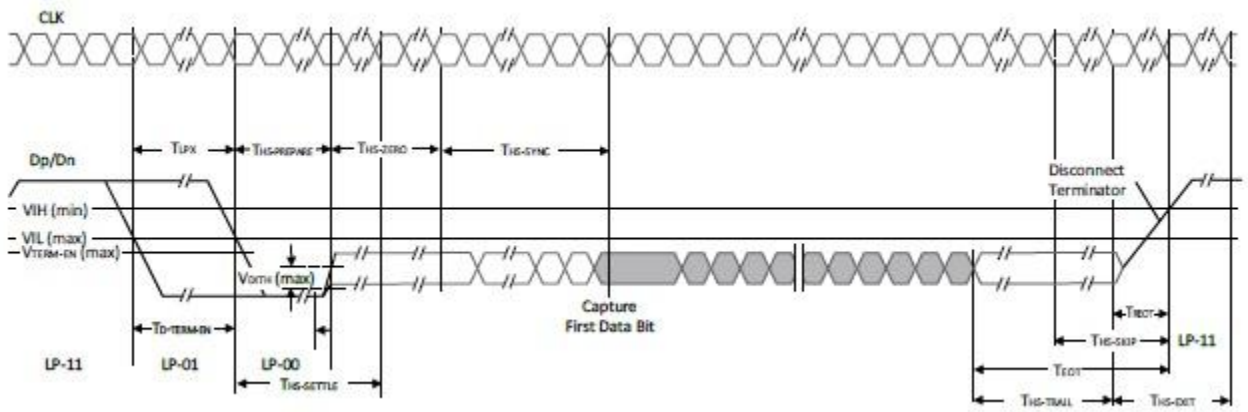

Figure 3.2. High Speed Data Transmission

The parallel transmit interface consists of clock, pixel data, and control signals. The pixel data width is configurable depending on the data type. The control signals are either data enable (DE), vertical and horizontal sync flags (VSYNC and HSYNC) for MIPI DSI applications, or frame valid and line valid for MIPI CSI-2 applications.

The clock is edge-aligned against data and control signals. All signal transitions happen in sync with the rising edge of pixel clock as shown in Figure 3.3 and Figure 3.4.

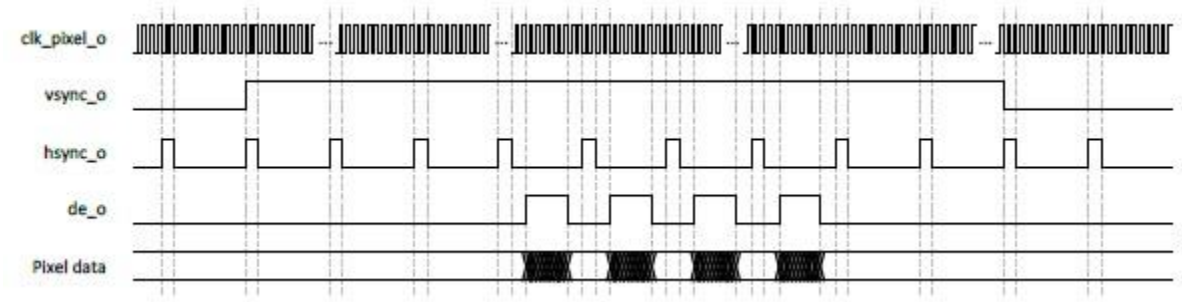

Figure 3.3. Parallel Transmit Interface Timing Diagram (DSI)

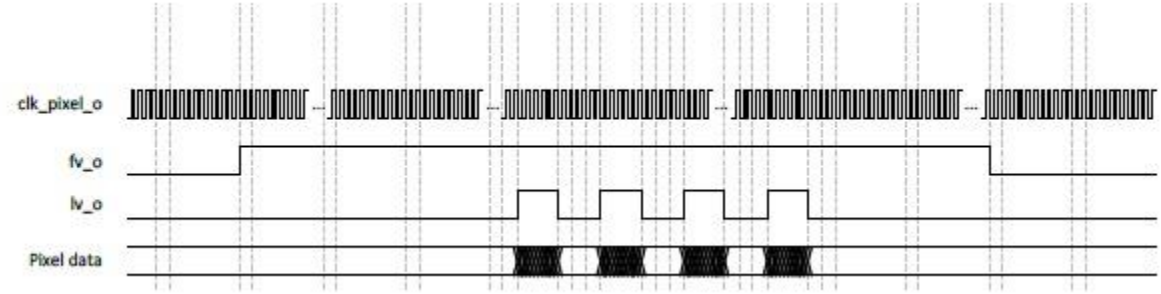

Figure 3.4. Parallel Transmit Interface Timing Diagram (CSI-2)

3.1. Design Components

The top-level design (*mipi2parallel.v*) consists of the following modules:

- rx_dphy
- b2p
- int_gpll

3.1.1. rx_dphy

This module must be created for the RX interface according to the required configuration, such as the number of lanes, bandwidth, and others. Refer to the CSI-2/DSI D-PHY Rx IP User Guide (FPGA-IPUG-02081) for details.

The following list provides the guidelines and parameter settings required for this reference design.

General Tab

- **RX Interface** – Selects CSI-2 or DSI.
- **D-PHY RX IP** – Selects Hard D-PHY or Soft D-PHY. The setting must match the RX_DPHY_HARD setting in *synthesis_directives.v*.
- **Number of RX Lanes** – Sets according to the RX interface configuration. The value must match the *NUM_RX_LANE_** setting in *synthesis_directives.v*.
- **RX Gear** – Selects 8 or 16. 16 is recommended when the RX byte clock speed exceeds 150 MHz or timing error occurs with Gear 8. 16 is supported for Hard D-PHY only.
- **Enable Deskew Calibration Detection** – Selects disabled (unchecked).
- **RX Line Rate** – Sets according to the RX interface configuration. 1,034 is the maximum for Soft D-PHY and 2,500 is the maximum for Hard D-PHY configuration.
- **D-PHY Clock Mode** – Selects Continuous or Non-continuous. Must match *RX_CLK_MODE_** setting (Continuous = HS_ONLY, Non-continuous = HS_LP) in *synthesis_directives.v*.

- **Sync Clock Frequency** – Sets to 100 MHz (default value).
- **Enable Lane Aligner Module (Soft D-PHY)** – Selects checkbox to enable (checked) for Soft D-PHY with 2-lane and 4-lane configurations.
- **CIL Bypass (Hard D-PHY)** – Selects disabled (unchecked).
- **Enable Packet Parser** – Selects checkbox to enable (checked).
- **Enable AXI4-Stream Interface** – Selects disabled (unchecked).
- **Enable LMMI Interface** – Selects disabled (unchecked).
- **Enable Miscellaneous Status Signals** – Selects enabled (checked).
- **Enable CRC Check** – Selects disabled (unchecked)
- **DSI Back-to-Back HS Packets (DSI)** – Sets to OFF.
- **Customize Data Settle Cycle** – Selects disabled (unchecked).
- **Configurable Data Settle Count** –Selects disabled (unchecked).

RX_FIFO Setting Tab

- **RX_FIFO Enable** – Selects disabled (unchecked) for continuous clock mode with Soft D-PHY, selects checkbox to enable (checked) for all other configurations.
- **Type** – Selects SINGLE for continuous clock mode with Hard D-PHY. Refer to the Non-Continuous Clock Mode section for details on the non-continuous clock mode.
- **Implementation** – Selects LUT for continuous clock mode with Hard D-PHY. Refer to the Non-Continuous Clock Mode section for details on the non-continuous clock mode.
- **Depth** – Selects 16 for continuous clock mode with Hard D-PHY. Refer to the Non-Continuous Clock Mode section for details on the non-continuous clock mode.
- **Default FIFO Read Delay** – Sets 1 for continuous clock mode with Hard D-PHY. Refer to the Non-Continuous Clock Mode section for details on the non-continuous clock mode.
- **Configurable FIFO Read Delay** – Selects disabled (unchecked).
- **Clock Mode** – Selects DC (dual clock).
- **Misc Signals** – Selects disabled (unchecked).

Soft PHY Tab

- **Delay Mode** – Selects Edge Clock Centered.
- **Enable Dynamic Delay Control** – Selects disabled (unchecked).

This module takes serial CSI-2/DSI data and outputs byte data after de-serialization in the MIPI High Speed mode. The *.ipx* file included in the project (*rx_dphy/rx_dphy.ipx*) can be used to reconfigure the IP per your configuration requirements. If you create this IP from scratch, it is recommended to set the design name to *rx_dphy* so that you do not need to modify the instance names of these IPs in *mipi2parallel.v* and the simulation setup file. Otherwise, you need to modify the names accordingly.

For four lanes, RX Gear = 16 in DSI mode, you may encounter static timing analysis (STA) failure in the Lattice Radiant software, especially in higher bandwidth. This failure occurs when there are two short and/or long packet headers within consecutive eight bytes in the MIPI DSI stream. DSI stream must be changed to increase the interval between two packet headers.

3.1.1.1. RX FIFO

RX FIFO is useful in non-continuous clock mode and when the continuous byte clock cannot have the exact same frequency as the non-continuous byte clock used in the D-PHY RX IP. RX FIFO resides after the word aligner for the Hard D-PHY RX IP and resides before the word aligner for the Soft D-PHY RX IP.

Hard D-PHY in Continuous Clock Mode

In this mode, the minimum configuration of RX FIFO is recommended (**Implementation** = LUT, **Depth** = 16, **Type** = SINGLE, **Packet Delay** = 1, **Clock Mode** = DC).

Soft D-PHY in Continuous Clock Mode

In this mode, RX FIFO is not necessary and **RX_FIFO Enable** must be disabled (unchecked).

Non-Continuous Clock Mode

In this mode, RX FIFO configuration depends on the relationship between the non-continuous byte clock in the D-PHY RX IP and the continuous byte clock. The continuous byte clock may be generated by GPLL. The non-continuous byte clock is used to write data to RX FIFO and the continuous byte clock is used to read data from RX FIFO.

- **Continuous Byte Clock = Non-Continuous Byte Clock**

  For this setting, the minimum configuration of RX FIFO is recommended (**Implementation** = LUT, **Depth** = 16, **Type** = SINGLE, **Packet Delay** = 1, **Clock Mode** = DC).

- **Continuous Byte Clock < Non-Continuous Byte Clock**

  For this setting, **Type** = SINGLE and **Packet Delay** = 1 are recommended, and other configurations depend on the frequency ratio between these two clocks. When the clock speed difference increases, the required depth of RX FIFO increases. You must know the horizontal blanking period of the incoming RX channel. For example, if one-line active video period is 40 µs and the horizontal blanking is 4 µs, we have 10% of extra time to process the active data. In this example, the continuous byte clock can be as slow as ~–10% compared to the non-continuous byte clock to avoid RX FIFO overflow.

- **Continuous Byte Clock > Non-Continuous Byte Clock**

  This setting has two options as follows:

  - Use **Type** = SINGLE with large **Packet Delay**

    Set the **Depth** with sufficient capacity to contain the necessary data to avoid RX FIFO underflow after FIFO read begins after the time specified by **Packet Delay**. In general, **Packet Delay** must be set close the depth of the RX FIFO. This configuration can be used when we have sufficient time interval between the last active line and the frame end short packet so that the frame end short packet is not written to RX FIFO while the frame still contains the last active line of video data.

  - **Use Type = QUEUE with Number of Queue Entries = 2**

    This is useful when the time interval between the last active line and frame end short packet is short or unknown. Depth must be set with sufficient capacity to contain one active line data and short packet data. This mode is also useful when line start and the line end short packets exist in the incoming RX stream. In this mode, **Number of Queue Entries** = 4 and extra depth is required for one line plus two short packet data. FIFO read begins after each HS data transaction is complete. EBR must be used. Counter Width is determined by the amount of one-line video data and extra overheads by preceding HS zero data and trail byte at the end of HS transmission.

- **Frequency relationship is unknown**

  When the continuous byte clock is within a certain range against the non-continuous byte clock (for example, two clocks come from different clock sources which have ppm tolerance), the clock frequency is unknown. To determine the clock, set **Type** = SINGLE and set **Packet Delay** to the midpoint of FIFO depth when the tolerance is at the ppm level. QUEUE can also be used as described above.

If you do not have detailed information regarding RX data (whether containing line start/end short packet, interval of the horizontal blanking period against active line period), set the continuous byte clock to be faster than the non-continuous byte clock and use **Type** = QUEUE with **Number of entries** = 4. This setting may require more EBR resources compared to **Type** = SINGLE.

3.1.2. b2p

This module must be created for the RX interface according to the required configuration, such as data type, the number of lanes, RX Gear, and others. Refer to the Byte-to-Pixel Converter IP User Guide (FPGA-IPUG-02079) for details.

The following list provides the guidelines and parameter settings required for this reference design.

- **Data Type** – Selects RGB888, RGB666, RAW8, RAW10, or RAW12 data type. The setting must match *synthesis_directives.v*.

- **RX Interface** – Selects DSI or CSI-2. Set the same type as the RX D-PHY IP. The setting must match *synthesis_directives.v*.
- **DSI Mode** – Selects Non-Burst Events or Burst (applicable to DSI Interface only).
- **Number of RX Lanes** – Selects 1, 2, or 4. Set the same value as the RX D-PHY IP.
- **RX Gear** – Selects 8 or 16. Set the same value as the RX D-PHY IP.
- **Byte Side Clock Frequency** – Sets the same value as obtained in the RX D-PHY IP.
- **Enable AXI4-Stream Receiver Interface** – Selects disabled (unchecked).
- **Number of Output Pixels Lanes** – Selects 1, 2, or 4.
- **Camera/Display Control Polarity** – Selects Positive.
- **DSI Sync Packet Delay (DSI)** – Uses the default value.
- **Pixel Clock Frequency** – Enters the appropriate value.
- **Pixel-Side Transmitter Interface** – Uses Native Interface.
- **Manual Adjust** – Selects disabled (unchecked).
- **FIFO Implementation** – Uses EBR.
- **Word Count** – Enters the appropriate value per the following equation:
  Word Count = (*NUM_PIXELS* × *PD_BUS_WIDTH*)/8
  For example: For 1080P with RAW10, the value is 1920 × 10/8 = 2400.
- **Enable Debug Ports** – Selects disabled (unchecked).
- **Register Interface** – Sets to OFF.

The Byte-to-Pixel Converter IP converts the D-PHY CSI-2/DSI standard based byte data stream to standard pixel data format. The *.ipx* file included in the project (*b2p/b2p.ipx*) can be used to reconfigure the IP per your configuration requirements. If you create this IP from scratch, it is recommended to set the design name to *b2p* so that you do not need to modify the instance name of this IP in the top-level design and in the simulation setup file. Otherwise, you need to modify the names accordingly.

3.1.3. int_gpll

You need to use the GPLL module to generate pixel clock for the pixel data, sync clock and continuous byte clock for non-continuous clock mode to be used for RX interface. Refer to the PLL Module User Guide (FPGA-IPUG-02063) for details on this IP.

The following list provides the default parameter settings for this reference design.

General Tab

- **Configuration Mode** – Selects frequency.
- **Enable Fractional-N Divider** – Selects enabled (checked).
- **CLKI: Frequency** – Sets to 100 MHz for the simulation. For hardware tests, you need to supply the input clock externally using on-board PLL or using an internal oscillator.
- **CLKOP: Frequency Desired Value** – Sets to 150 MHz. Provides a continuous byte clock for non-continuous RX clock mode. You need to modify this value according to your requirements.
- **CLKOS: Frequency Desired Value** – Sets to 100 MHz. Provides pixel clock to the Byte-to-Pixel IP. You need to modify this value according to your requirements.
- **CLKOS2: Frequency Desired Value** – Sets to 100 MHz. Used as a sync clock for RX D-PHY.

Optional Ports Tab

- **Provide PLL Reset** – Selects enabled (checked).
- **Provide PLL Lock Signal** – Selects enabled (checked).

The *.ipx* file included in the project (*int_gpll/int_gpll.ipx*) can be used to reconfigure the IP per your configuration requirements. If you create this IP from scratch, it is recommended to set the design name to *int_gpll* so that you do not need to modify the instance names of these IPs in the *mipi2parallel.v* file. Otherwise, you need to modify the name accordingly.

3.2. Clocking Scheme

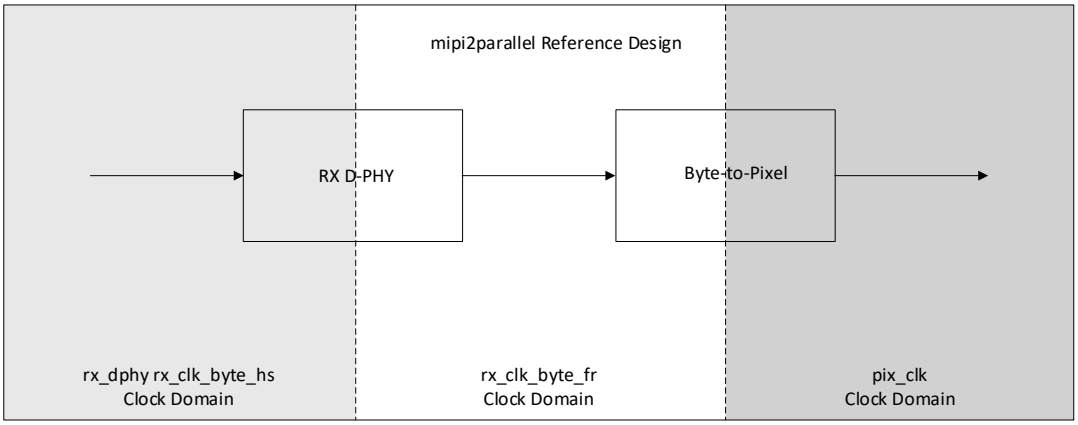

Figure 3.5. Reference Design Clock Domains Block Diagram

Figure 3.5 shows the clock domains of the data path of the designs.

- RX D-PHY write byte clock (rx_clk_byte_hs) captures the D-PHY byte data into a built-in FIFO of the rx_dphy. This clock is generated by dividing the D-PHY clock lanes by the number of gears.
- RX D-PHY free running byte clock (rx_clk_byte_fr) reads data from the built-in FIFO of the rx_dphy.
  - In continuous clock mode, this clock is the same as rx_dphy rx_clk_byte_hs.
  - In non-continuous clock mode, this clock comes from CLKOP of the int_gpll module.
- The pixel clock domain (pix_clk) is used when the data is already in pixel format. Sync signals also use this domain. This clock is generated by CLKOS of the int_gpll module.

3.3. Reset Scheme

The system level reset is routed to the reset_n_i pin of the top-level module as an active-low reset. Asserting this reset asynchronously resets all components in the design. After the system reset is released, all design components are out of reset. All the reset signals are asynchronously asserted and synchronously deasserted with the respective clock domains.

4. Reference Design Parameter Description

This reference design has the following directive files:

- *synthesis_directives.v* – Used for design compilation by the Lattice Radiant software and for simulation.
- *simulation_directives.v* – Used for simulation.

You can modify these directive files according to your own configuration. The settings in these files must match CSI-2/DSI D-PHY Rx IP, Byte-to-Pixel Converter IP, and GPLL IP settings created by the Lattice Radiant software.

4.1. Synthesis Directives

Table 4.1 shows the synthesis directives that affect this reference design. These are used for both synthesis and simulation. As shown in Table 4.1 and Table 4.2, some parameter selections are restricted by other parameter settings.

Table 4.1. Synthesis Directives

Category	Directive	Description
Debug interface	DPHY_DEBUG_ON	Enables additional debug interface in the design.
RX Interface	RX_TYPE_DSI	Specifies the D-PHY Receive interface. Defines only one of the directives.
	RX_TYPE_CSI2	
Number of RX Lanes	NUM_RX_LANE_1	Specifies the number of lanes in the RX interface. Defines only one of the three directives.
	NUM_RX_LANE_2	
	NUM_RX_LANE_4	
Number of Pixel Per Clock	NUM_PIX_LANE_1	Defines the number of pixel lanes or pixel per clock. Note that not all modes are available for a combination of data types and number of lanes.
	NUM_PIX_LANE_2	
	NUM_PIX_LANE_4	
RX D-PHY Clock Gear	RX_GEAR_8	Specifies the RX D-PHY clock gear. Selects only one of the directives.
	RX_GEAR_16	
Sync Signal Polarity	SYNC_POLARITY_POS	Selects the sync signal (VSYNC, HSYNC) polarity. Defines only one of the directives. Applicable only to DSI.
	SYNC_POLARITY_NEG	
RX D-PHY Clock Mode1	RX_CLK_MODE_HS_ONLY	Specifies the RX D-PHY clock mode. Defines only one of the directives. HS_LP mode means non-continuous clock mode and HS_ONLY mode means continuous clock mode. HS_LP mode works only if RX byte clock for the corresponding RX channel can be generated internally or directly fed from the I/O pin.
	RX_CLK_MODE_HS_LP	
RX D-PHY type	RX_DPHY_HARD	Specifies the Hard D-PHY implementation. If RX D-PHY type is not specified, Soft D-PHY implementation is used.
RX Data Type	DT_RGB888	Specifies the data type on the D-PHY RX interface. Defines only one of the directives.
	DT_RGB666	
	DT_RAW8	
	DT_RAW10	
	DT_RAW12	

4.2. Simulation Directives

Table 4.2 shows the simulation directives for this reference design.

Table 4.2. Simulation Directives

Category	Directive	Description
Video Data Configuration on RX Interface	NUM_FRAMES {value}	Number of video frames fed by the testbench.
	NUM_LINES {value}	Number of active lines per frame.
	NUM_PIXELS {value}	Number of active video pixels per line.
Blanking Mode	LP_BLANKING	Enables the low power blanking during vertical blanking period. If not defined, high-speed blanking is used by default.1
Sync Type of Video Data	NON_BURST_SYNC_EVENTS	Enables Non-Burst Sync Event mode.[1]
End of Transmission Packet	EOTP	When enables, DSI model generates EOTP packet.[1]
RX D-PHY Clock Period	DPHY_CLK {value}	RX D-PHY clock period in ps.
GPLL Reference Clock Period	REF_CLK {value}	GPLL reference input clock period in the mipi2parallel module.
DSI Sync Checker	VSYNC_HSYNC_CHECK	Enables sync checker in testbench.[1]
Vertical Sync Width	VSYNC_WIDTH {value}	Vertical SYNC width. This value must match number of HSYNC pulses inside VSYNC active region setting of the Byte-to-Pixel Converter IP.[1]
Horizontal Sync Width	HSYNC_WIDTH {value}	Horizontal SYNC width. This value must match the number of pix clock cycles HSYNC remains active setting of the Byte-to-Pixel Converter IP.[1]
Sync Error Messages	SIM_STOP_AT_HSYNC_VSYNC_WIDTH_FAIL {value}	Simulation stops with error messages when Hsync or Vsync failure is detected. Set value to either 1 or 0.[1]

Note:

1. Applicable to DSI Mode only.

5. Signal Description

Table 5.1 shows the top-level I/O of this reference design. Actual I/O depend on your configurations. All necessary I/O ports are automatically declared by compiler directives.

Table 5.1. MIPI-to-Parallel Top-Level I/O

Port Name	Direction	Description
Clocks and Resets		
ref_clk_i	I	Input reference clock for the GPLL.
reset_n_i	I	Asynchronous active low system reset.
MIPI D-PHY RX Interface		
clk_p_i	I/O	Positive differential RX D-PHY input clock.
clk_n_i	I/O	Negative differential RX D-PHY input clock.
d_p_io[BUS_WIDTH – 1:0][1]	I/O	Positive differential RX D-PHY input data.
d_n_io[BUS_WIDTH – 1:0][1]	I/O	Negative differential RX D-PHY input data.
Parallel Interface		
clk_pixel_o	O	Pixel clock generated from internal GPLL.
vsync_o	O	Vertical sync indicator (active high/low). Goes high/low when VSYNC start short packet is received. Goes low/high when VSYNC end short packet is received. Available only for MIPI DSI mode.[2]
hsync_o	O	Horizontal sync indicator (active high/low). Goes high when either VSYNC/HSYNC start or VSYNC end short packet is received. Goes low when HSYNC end short packet is received. Available only for MIPI DSI mode.[3]
de_o	O	Data enable indicator (active high). Goes high at the start of valid pixel data and goes low at the end of valid pixel data. Available only for MIPI DSI mode.
fv_o	O	Frame valid indicator (active high). Goes high when frame start short packet is received and goes low when frame end short packet is received. Available only for MIPI CSI-2 mode.
lv_o	O	Line valid indicator (active high). Goes high at the start of valid pixel data and goes low at the end of valid pixel data. Available only for MIPI CSI-2 mode.
pd_o[PD_BUS_WIDTH * NUM_PIX_LANE -1:0][4,5]	O	Pixel data output. Data width depends on the selected data type and pixel per clock configurations.
p_odd_o[1:0]	O	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. 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
pd_blue_o[(BITS_PER_COLOR[6] * NUM_PIX_LANE)-1:0]	O	Blue pixel data output. Applicable when the Byte-to-Pixel Converter IP uses RGB888 or RGB666 data type.
pd_green_o[BITS_PER_COLOR[6] * NUM_PIX_LANE)-1:0]	O	Green pixel data output. Applicable when the Byte-to-Pixel Converter IP uses RGB888 or RGB666 data type.
pd_red_o[BITS_PER_COLOR[6] * NUM_PIX_LANE)-1:0]	O	Red pixel data output. Applicable when the Byte-to-Pixel Converter IP uses RGB888 or RGB666 data type.
Debug Interface		
payload_o[BUS_WIDTH * RX_GEAR-1:0]	O	Payload data output of D-PHY RX. Enables when DPHY_DEBUG_ON is enabled in *synthesis_directives.v*.
payload_en_o	O	Valid payload data output of D-PHY RX. Enables when DPHY_DEBUG_ON is enabled in *synthesis_directives.v*.
rx_clk_byte_fr_o	O	Continuous byte clock generated by GPLL or D-PHY RX. Enables when DPHY_DEBUG_ON is enabled in *synthesis_directives.v*.

Port Name	Direction	Description
lp_av_en_o	O	Asserts when long packet is the same as ref_dt. Enables when DPHY_DEBUG_ON is enabled in *synthesis_directives.v*.
wc_o [15:0]	O	Indicates word count detected in payload data. Enables when DPHY_DEBUG_ON is enabled in *synthesis_directives.v*.

Notes:

1. BUS_WIDTH – Number of D-PHY Lanes 1, 2, or 4 (available on the user interface).
2. For Non-Burst Sync Events, VSYNC end short packet is not received, and vsync_o goes high/low as specified in VSYNC_WIDTH in *simulation_directives.v*.
3. For Non-Burst Sync Events, HSYNC end short packet is not received, and hsync_o goes high/low as specified in HSYNC_WIDTH in *simulation_directives.v*.
4. PD_BUS_WIDTH – The value can be 8, 10, 12, 18, or 24 according to the selected data type RAW8, RAW10, RAW12, RGB666, and RGB888 respectively. The value can be 18 or 24 for RGB666 or RGB888 respectively for DSI. The value can be 8, 10, 12, or 24 for RAW8, RAW10, RAW12, or RGB888 respectively for CSI-2.
5. NUM_PIX_LANE – Number of pixels per clock as defined in *synthesis_directives.v*.
6. BITS_PER_COLOR – The value is 6 BPC for RGB666 and 8 BPC for RGB888.

6. Running the Reference Design

This section describes how to run the MIPI-to-Parallel reference design using the Lattice Radiant software. For more details on the Lattice Radiant software, refer to the Lattice Radiant Software User Guide.

6.1. Opening the Reference Design Project

To open the reference design main project file, follow these steps:

1. Open the Lattice Radiant software.
2. Click **File > Open Project** and from the project database, open the Lattice Radiant software project file (*.rdf*) from the *<design_directory>/fpga_lifcl/radiant* directory. The reference design project opens.

6.2. Compiling and Generating the Bitstream File

This section provides the procedure of creating your FPGA bitstream file using the Lattice Radiant software. Note that this reference design uses virtual I/O pinout (defined in *.pdc* file) and you need to disable this virtual pinout to perform the Lattice Radiant compilation.

To compile and create a new FPGA bitstream file using the Lattice Radiant software, follow these steps:

1. Open the Lattice Radiant software and reference design project. Refer to the Opening the Reference Design Project section.
2. Click **Export Files** to generate the bit file. View the log message in the Export Reports folder for the generated bitstream.
3. Locate the new bitstream file in the *impl1* directory.

7. Simulating the Reference Design

To simulate the design, perform the following steps:

1. Unzip the reference design *.zip* file.
2. Open the reference design project file (*mipi2parallel.rdf*) using the Lattice Radiant software. Refer to the Opening the Reference Design Project section.
3. To modify the stimulus sent (number of pixels, number of lines, and other data), you can modify the compiler directives manually. For details of the directives, refer to Table 4.1 and Table 4.2.
4. Open QuestaSim® simulator in the Lattice Radiant software.
5. Navigate to the *<design_directory>/fpga_lifcl/sim/questa* directory using the QuestaSim transcript.
6. Run the script using the following command:
```
VSIM 12> do run_sim.do
```

7.1. Simulation Results

By executing the script in QuestaSim, compilation and simulation are executed automatically. The testbench takes all data comparison between the expected data and output data from the reference design.

```
# 392374127000 Total number of bytes of active pixels per line = 5760
#
# 392374127000 Total number of active lines = 3
#
# Generate data for line 0
#
# Generate data for line 1
#
# Generate data for line 2
#
# 392995121000 SoT started...
#
# 393000614000 SoT ended...
#
##### 393000614000 Transmit DPHY packets ongoing... #####
##### HSYNC assertion #####
##### HSYNC de-assertion #####
##### DE assertion #####
##### DE de-assertion #####
##### HSYNC assertion #####
##### HSYNC de-assertion #####
##### DE assertion #####
##### DE de-assertion #####
##### HSYNC assertion #####
##### HSYNC de-assertion #####
##### DE assertion #####
##### DE de-assertion #####
##### 459440694000 Transmit DPHY packets DONE! #####
# 459440694000 HS trail data lane0 started...
# 459440694000 HS trail data lane1 started...
# 459529858000 HS trail data lane0 ended...
# 459529858000 HS trail data lane1 ended...
##### HSYNC assertion #####
##### HSYNC de-assertion #####
# 480979102000 SoT started...
#
# 480984595000 SoT ended...
#
```

```
##### 480984595000 Transmit DPHY packets ongoing... #####
##### HSYNC assertion #####
##### 481437747000 Transmit DPHY packets DONE! #####
# 481437747000 HS trail data lane0 started...
# 481437747000 HS trail data lane1 started...
# 481526911000 HS trail data lane0 ended...
# 481526911000 HS trail data lane1 ended...
##### HSYNC de-assertion #####
# 502976155000 SoT started...
#
# 502981648000 SoT ended...
#
##### 502981648000 Transmit DPHY packets ongoing... #####
##### HSYNC assertion #####
##### 503434800000 Transmit DPHY packets DONE! #####
# 503434800000 HS trail data lane0 started...
# 503434800000 HS trail data lane1 started...
# 503523964000 HS trail data lane0 ended...
# 503523964000 HS trail data lane1 ended...
##### HSYNC de-assertion #####
# 524973208000 SoT started...
#
# 524978701000 SoT ended...
#
##### 524978701000 Transmit DPHY packets ongoing... #####
##### HSYNC assertion #####
##### 525431853000 Transmit DPHY packets DONE! #####
# 525431853000 HS trail data lane0 started...
# 525431853000 HS trail data lane1 started...
# 525521017000 HS trail data lane0 ended...
# 525521017000 HS trail data lane1 ended...
##### HSYNC de-assertion #####
# 546349267000 FRAME #1 ended...
#
# 546349267000 FRAME #2 started...
#
# 546970261000 SoT started...
#
# 546975754000 SoT ended...
#

When the simulation is finished, the following statements are displayed

# ------------------------------------------
# **** I N F O : Byte Count is 51840
# **** I N F O : NUM_FRAMES=3, NUM_LINES=3, TOTAL BYTES =51840
# ------------------------------------------
# ------------------------------------------------------
# ----------------- SIMULATION PASSED ----------------
# ------------------------------------------------------
# 1338334713000 TEST END
```

The simulation waveform can be accessed by opening the *vsim.wlf* file in QuestaSim from the simulation directory. The following waveforms show the simulation waveforms of the full view of three lines and three frames for the DSI and CSI-2 interfaces. The waveforms show all the top-level I/O and other signals.

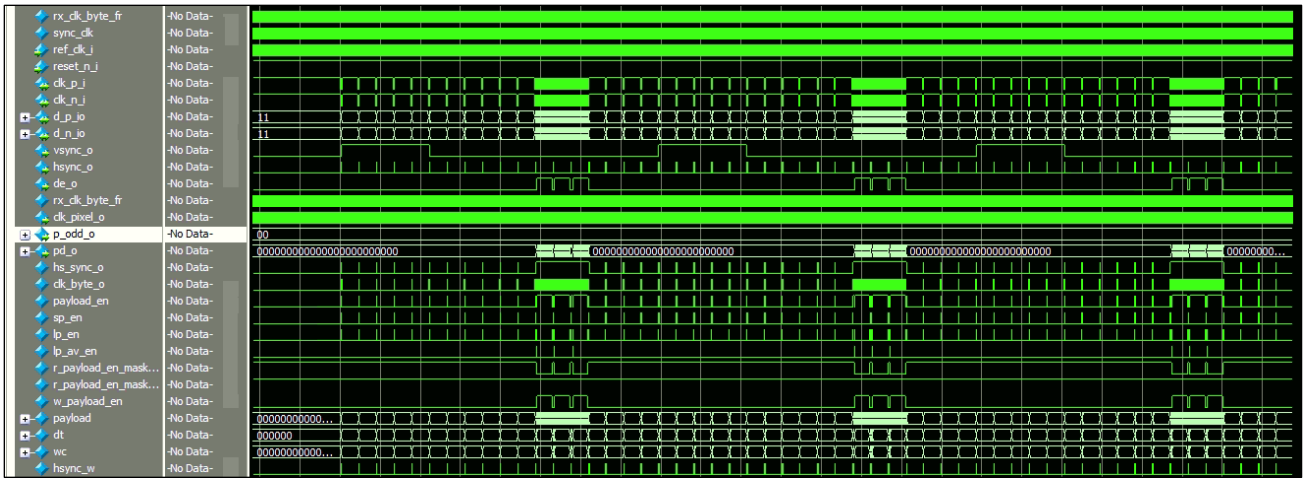

Figure 7.1. Simulation Waveform for DSI Interface

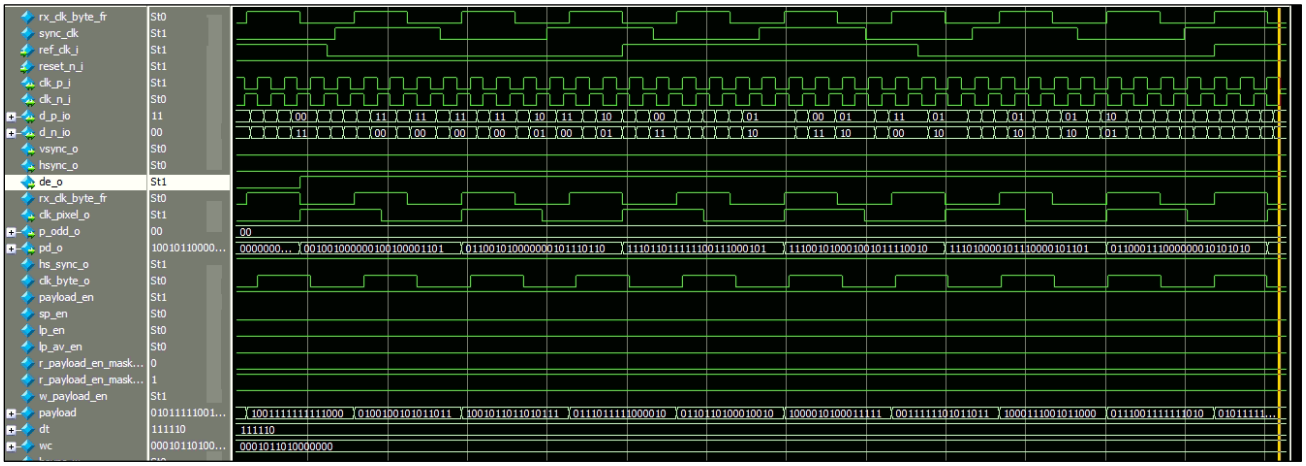

Figure 7.2. Simulation Waveform for DSI Interface – Zoom View

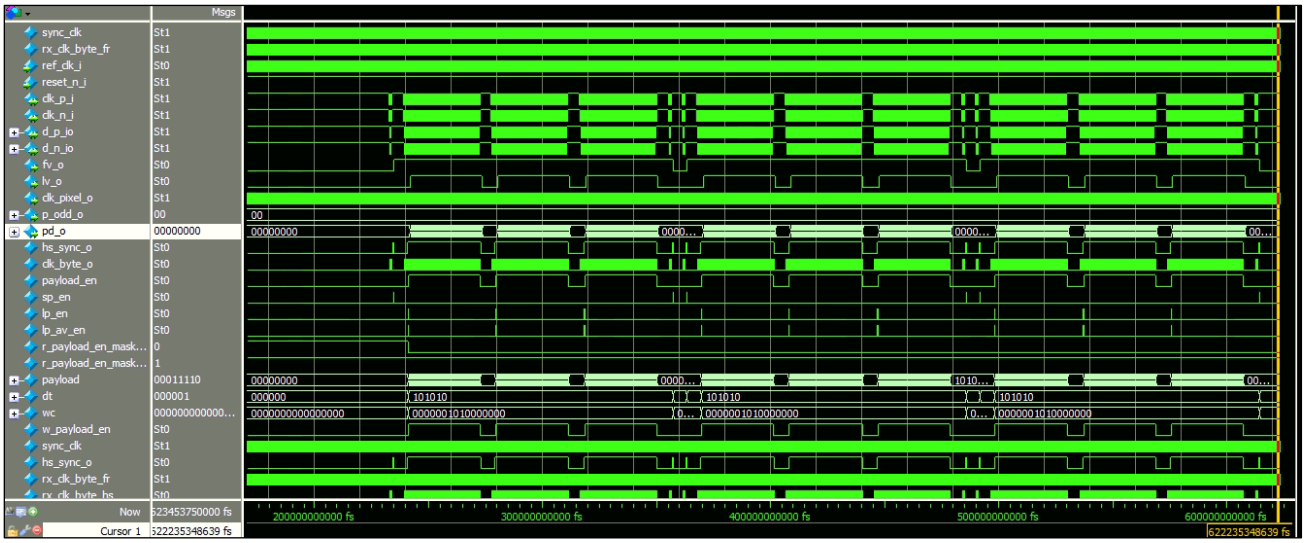

Figure 7.3. Simulation Waveform for CSI-2 Interface

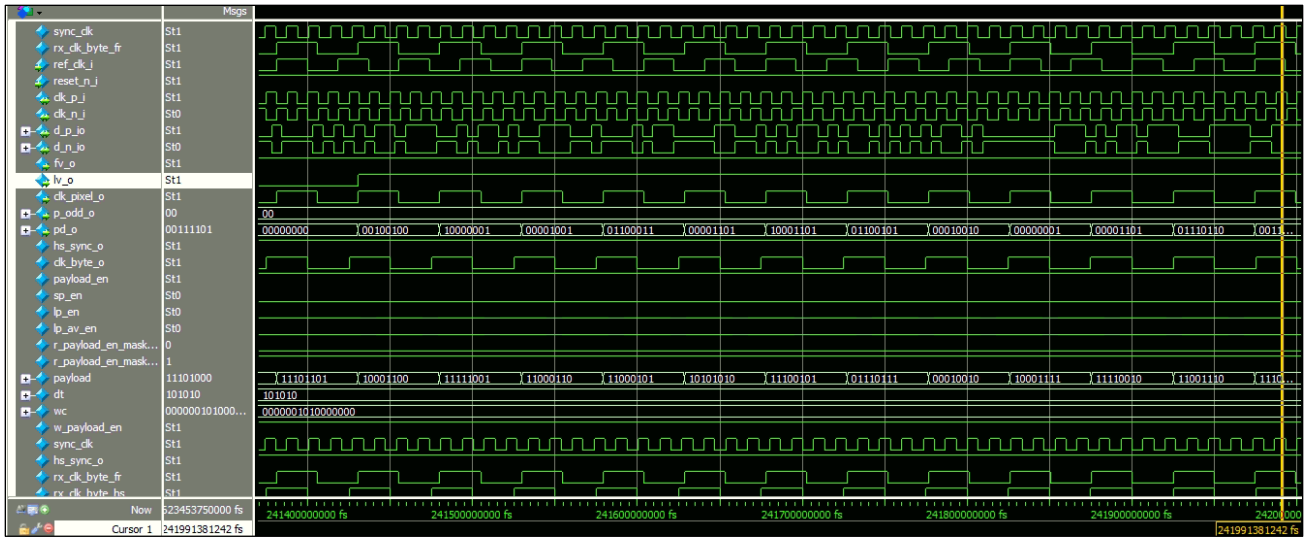

Figure 7.4. Simulation Waveform for CSI-2 Interface – Zoom View

8. Customizing the Reference Design

8.1. Using the Excel Calculator Helper

The reference design includes a simple calculator to determine the D-PHY clock, byte clock, and pixel clock frequencies required by the CSI-2/DSI D-PHY RX IP and Byte-to-Pixel Converter IP. The calculator can be found in the *<design_directory>/misc* directory.

Items	Values	Notes
Line Rate (Mbps)	1200	Must be between 160 and 2500Mbps
Number of Lanes	1	1, 2 or 4 lanes
Number of Gear	8	Select 8 or 16 Gear
Interface	CSI-2	Select CSI-2 or DSI
Data Type	RGB888	
Pixel Per Clock	1	1, 2, or 4 PPC
Number of bits per pixel clock	24	
D_PHY clock (MHz)	600.0000	D-PHY serial clock
Pixel Clock Freq (MHz)	50.0000	int_gpll's CLKOS frequency.
Byte_clk_fr (MHz)	150.0000	int_gpll's CLKOP frequency, used by RX DPHY as a read clock. Only used by non-continuous clock mode. For continuous clock mode, RX DPHY uses write clock as the read clock as well

LEGEND	
must be filled by user	

Figure 8.1. Excel Calculator Tool

8.2. Customization Flow

To customize this reference design, follow these steps:

1. Modify *synthesis_directives.v* and *simulation_directives.v* according to your design requirements, such as the number of D-PHY lanes, protocol, video data type, and other settings.
2. Ensure the parameters defined in the Byte-to-Pixel Converter IP match the parameters in *synthesis_directives.v*. Click **Generate** after you update the IP parameters in the Lattice Radiant software.
3. Verify that the CSI-2/DSI D-PHY Rx IP uses the same number of lanes, interface type, and clock mode as specified in *synthesis_directives.v*. Click **Generate** after you update the IP parameters in the Lattice Radiant software.
4. Update the pixel clock, byte clock, and sync clock of the reference design by modifying the PLL IP parameters accordingly. You can use the Excel Calculator tool included with this reference design to assist with the calculations.
5. Run the Lattice Radiant software compilation or simulation as described in the Running the Reference Design and Simulating the Reference Design sections.

9. Resource Utilization

Resource utilization depends on the configuration used. The resource utilization example is a reference. The actual usage varies, especially when EBR usage depends on the horizontal resolution of each RX channels and clock frequency relationship between non-continuous byte clock and continuous byte clock when non-continuous clock mode is used.

Table 9.1. Resource Utilization Examples for the LIFCL-40 Device

Configuration	LUT4	FPU Register	EBR	I/O Buffer
4-lane, Gear 16, Hard D-PHY, DSI, RGB888, 4 pixel/clock	1,377	1,347	11	3
1-lane, Gear 8, Hard D-PHY, DSI, RGB888, 1 pixel/clock	864	588	2	3
4-lane, Gear 8, Hard D-PHY, CSI-2, RAW10, 1 pixel/clock	930	720	6	3
1-lane, Gear 8, Soft D-PHY, CSI-2, RAW8, 1 pixel/clock	783	580	3	8

10. Debugging

You can use the tool below to debug MIPI-to-Parallel reference design issues.

10.1. Reveal Analyzer

The Reveal™ Analyzer continuously monitors signals within the FPGA for specific conditions that range from simple to complex conditions. When the trigger condition occurs, the Reveal Analyzer saves signal values preceding, during, and following the event for analysis, including a waveform presentation. The data can be saved in the following format:

- Value change dump file (*.vcd*) that can be used with tools such as QuestaSim.
- ASCII tabular format that can be used with tools such as Microsoft® Excel.

Before running the Reveal Analyzer, use the Reveal Inserter to add Reveal modules to your design. In these modules, specify the signals to monitor, define the trigger conditions, and other preferred options. The Reveal Analyzer supports multiple logic analyzer cores using hard or soft JTAG interface. You can have up to 15 modules, typically one for each clock region of interest. When the modules are set up, regenerate the bitstream data file to program the FPGA.

During debug cycles, this tool uses a divide and conquer method to narrow down the problem areas into many small functional blocks to control and monitor the status of each block.

Refer to the Reveal User Guide for Radiant Software for details on how to use the Reveal Analyzer. Refer to the simulation *wave.do* file from the Simulating the Reference Design section for the critical signals to be viewed with the Reveal Analyzer.

11. Known Limitations

This reference design has the following limitations:

- The MIPI DSI interface supports only the following data types: RGB888, RGB666.
- The MIPI CSI-2 interface supports only the following data types: RGB888, RAW8, RAW10, RAW12.
- Timing violation may occur in higher bandwidth in DSI with four lanes, Gear = 16.

References

- MIPI Alliance web page for D-PHY Version 1.2, Display Serial Interface 2 (DSI) Version 1.2, and Camera Serial Interface 2 (CSI-2) Version 1.2 and Version 2.0
- CSI-2/DSI D-PHY Rx IP User Guide (FPGA-IPUG-02081)
- Byte-to-Pixel Converter IP User Guide (FPGA-IPUG-02079)
- PLL Module User Guide (FPGA-IPUG-02063)
- Reveal User Guide for Radiant Software
- CrossLink-NX web page
- MIPI DSI/CSI-2 to Parallel Bridge Reference Design web page
- Lattice Radiant Software web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

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

Revision 1.2, September 2025

Section	Change Summary
All	Performed minor formatting and editorial edits.
Inclusive Language	Added inclusive language boilerplate.
Abbreviations in This Document	Updated list of abbreviations.
Introduction	• Reworked section 1 Introduction. • Reworked section 1.1 *Supported Device and IP* and renamed to section 1.1 Quick Facts. • Reworked section 1.2 Features List. • Reworked section 1.5 *Conventions* and renamed to section 1.3 Naming Conventions.
Directory Structure and Files	Reworked section 7 *Design Package and Project Setup* and renamed to section 2 Directory Structure and Files.

Functional Description	• Reworked section 1.3 *Block Diagram* and section 1.4 *Functional Description*, and renamed to section 3 Functional Description.
	• Reworked section 3 *Design and Module Description* and section 4 *Design and File Modifications*, and renamed to section 3.1 Design Components.
	• Added the following sections:
	  • 3.2 Clocking Scheme
	  • 3.3 Reset Scheme

Reference Design Parameter Description	• Reworked section 2 *Parameters and Port List* and renamed to 4 Reference Design Parameter Description. • Reworked section 2.1 *Synthesis Directives* and moved to 4.1 Synthesis Directives. • Reworked section 2.2 *Simulation Directives* and moved to 4.2 Simulation Directives.
Signal Description	Reworked section 2.3 *Top-Level I/O* and renamed to section 5 Signal Description.
Running the Reference Design	Added this section.
Simulating the Reference Design	Reworked section 5 *Design Simulation* and renamed to section 7 Simulating the Reference Design.
Customizing the Reference Design	Added this section.
Resource Utilization	Reworked section 8 *Resource Utilization* and moved to section 9 Resource Utilization.
Debugging	Added this section.
Known Limitations	Reworked section 6 *Known Limitations* and moved to section 11 Known Limitations.
References	Updated references.

Revision 1.1, July 2024

Section	Change Summary
All	• Minor adjustments in formatting across the document. • Changed document name from MIPI to Parallel with CrossLink-NX to *MIPI DSI/CSI-2 to Parallel Bridge*.
Introduction	• Updated Radiant version and IP version for LIFCL-40 and LIFCL-17 and added LIFCL-33 device support in Table 1.1. Supported Device and IP and moved Supported Device and IP to this section. • Updated MIPI D-PHY Specification version from 1.1 to *1.2* in Functional Description.
Design and Module Description	• Updated Figure 3.1. rx_dphy IP Creation in Lattice Radiant including figure caption and removed rx_dphy IP Creation in Lattice Radiant (2/2) figure. • Updated maximum values of Soft D-PHY and Hard D-PHY to *1034* and *1500* respectively in RX Line Rate bullet point. • Updated Figure 3.2. b2p IP Creation in Lattice Radiant, Figure 3.3. int_osc IP Creation in Lattice Radiant, and Figure 3.4. int_gpll IP Creation in Lattice Radiant.
Design and File Modifications	Updated RX D-PHY IP version to 1.5.0 and Byte2Pixel IP version to 1.6.0.
References	Added reference to CrossLink-NX and Lattice Radiant web pages.

Section	Change Summary
Technical Support Assistance	Added reference to the Lattice Answer Database on the Lattice website.

Revision 1.0, February 2021

Section	Change Summary
All	Initial release.

www.latticesemi.com