CrossLink Parallel-to-MIPI User Guide

Reference Design

FPGA-RD-02293-1.0

August 2024

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.

Abbreviation	Definition
CSI-2	Camera Serial Interface 2
DPI	Display Pixel Interface
DSI	Display Serial Interface
EBR	Embedded Block RAM
ECC	Error Correction Code
HS	High Speed
LP	Low Power
LUT	Look Up Table
MIPI	Mobile Industry Processor Interface
PLL	Phase Locked Loop
P2B	Pixel2Byte
TX	Transmitter

1. Introduction

The Mobile Industry Processor Interface (MIPI®) D-PHY was developed primarily to support camera and display interconnections in mobile devices, and MIPI D-PHY has become the primary high-speed PHY solution in industry 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. The interface meets the demanding requirements of low power, low noise generation, and high noise immunity that mobile phone designs demand.

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

The Lattice Semiconductor Parallel-to-MIPI reference design allows the quick interface for a processor with an RGB interface to a display with a MIPI DSI interface, or a camera with a CMOS interface to a processor with CSI-2 interface. This reference design provides the conversion for CrossLink™ devices and 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 Lattice reference design web page: Parallel to MIPI CSI-2 / DSI Display Interface Bridge Reference Design

Table 1.1. Summary of the Reference Design

General	Target devices	LIF-MD6000
	Source code format	Verilog
Simulation	Functional simulation	Performed
	Timing simulation	Performed
	Testbench	Available
	Testbench format	Verilog
Software Requirements	Software tool and version	Lattice Diamond™ software version 3.13
	IP version	CSI-2/DSI D-PHY Transmitter Submodule IP v1.4 Pixel-to-Byte Converter IP v1.3
Hardware Requirements	Board	LIFMD6000 Master Link Board
	Cable	USB-to-mini-USB cable for programming purposes

1.2. Features

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

- Compliant with MIPI D-PHY v1.1, MIPI DSI v1.1, and MIPI CSI-2 v1.1 specifications
- Supports MIPI DSI and MIPI CSI-2 interfacing up to 6 Gb/s
- Supports 1, 2, or 4 MIPI D-PHY data lanes
- Supports non-burst mode with sync events for transmission of DSI packets only
- Supports low power (LP) mode during vertical and horizontal blanking
- Supports common MIPI DSI compatible video formats (RGB888, RGB666)
- Supports common MIPI CSI-2 compatible video formats (RGB888, RAW8, RAW10, RAW12)

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 (asserted when value is logic 0)
- _*i* are input signals
- _*o* are output signals
- _*io* are bidirectional signals

1.3.3. Data Ordering and Data Types

The highest bit within a data bus is the most significant bit. 8-bit parallel data is serialized to 1-bit data stream on each MIPI D-PHY data lane where bit 0 is the first transmitted bit.

Table 1.2. Pixel Data Order from the Core Module

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

2. Directory Structure and Files

```
.
└── lifmd_p2m_rd/
    ├── lifmd_p2m.ldf                       - Lattice Diamond design file
    ├── doc/                                - Contains IP configuration helper calculator
    │   └── p2m_helper_calculator.xlsx
    ├── impl1/                              - Contains Lattice Diamond implementation files and a pre-compiled bitstream file
    │   ├── lifmd_p2m_impl1.bit
    │   └── <other Diamond generated files>
    ├── src/                                - Contains verilog top level design, modules and synthesis directive files for parallel2MIPI design
    │   ├── p2m_top.v
    │   ├── p2m_csi2_glue.v
    │   ├── p2m_dsi_glue.v
    │   └── synthesis_directives.v
    ├── ip/                                 - Contains MIPI TX DPHY and Pixel-to-byte IP files used in the design
    │   ├── p2b
    │   ├── tx_dphy
    │   └── ip.sbx
    ├── simulation/                         - Contains simulation log and script files
    │   ├── parallel2mipi_msim.do
    │   ├── dphy_checker_timing.log
    │   ├── input_data.log
    │   ├── output_data.log
    │   └── simulation.log
    └── testbench/                          - Contains simulation directives and other testbench files
        ├── verilog
        ├── parallel2mipi_tb.v
        ├── simulation_directives.v
        ├── dut_inst.v
        ├── dphy_checker.v
        └── vid_timing_gen_driver.v/
            └── tb_include/
                └── tb_params.v
```

Figure 2.1. Directory Structure of the CrossLink Parallel-to-MIPI Reference Design

3. Functional Description

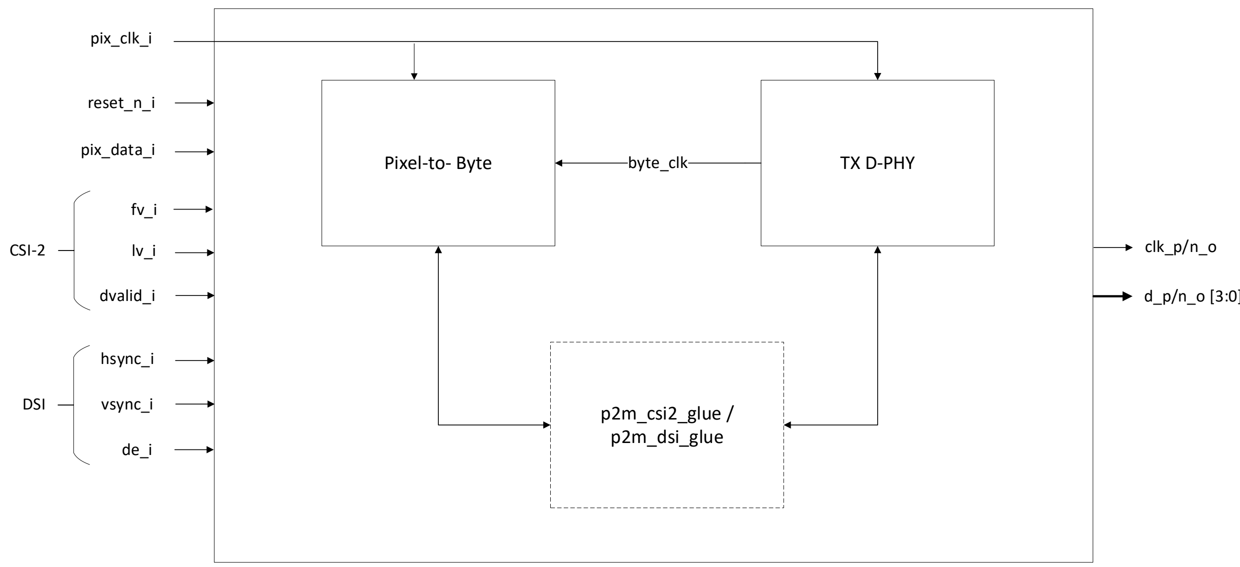

Figure 3.1. Parallel-to-MIPI Reference Design Block Diagram

The block level diagram of the Parallel-to-MIPI reference design mainly consists of the Pixel-to-Byte IP, TX D-PHY IP, and p2m_csi2/dsi_glue module. Depending on the selected TX D-PHY interface in the synthesis directives, the p2m_csi2_glue or p2m_dsi_glue module is instantiated to drive the assertion of byte data, header information, and other control signals to the TX-DPHY.

The Parallel-to-MIPI reference design converts a standard parallel video interface into either DSI or CSI-2 byte packets. The input interface for the design consists of a pixel bus (RGB888, RGB666), vertical and horizontal sync flags, a data enable and a clock for DSI and pixel bus (RGB888, RAW8, RAW10, and RAW12), frame and line valid flags, and a clock for CSI-2.

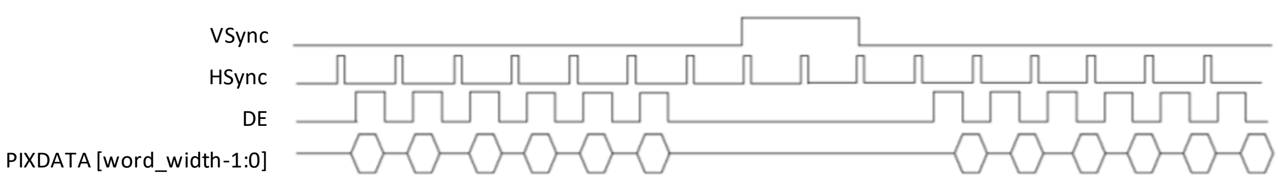

Figure 3.2. Display Parallel Input Bus Waveform

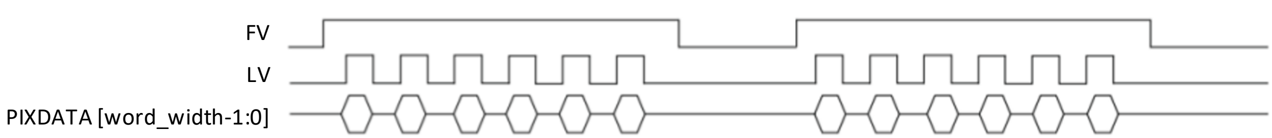

Figure 3.3. Camera Sensor Parallel Input Bus Waveform

This parallel bus in the above waveforms is converted to the appropriate DSI or CSI-2 output format. The DSI/CSI-2 output serializes HS (High Speed) data and controls LP (Low Power) data and transfers the data through the MIPI D-PHY IP. MIPI D-PHY also has a maximum of 5 lanes per channel which are consists of one clock lane and up to 4 data lanes. The maximum D-PHY data rate per lane is 1.5 Gb/s.

3.1. Design Components

The CrossLink Parallel-to-MIPI reference design top-level consists of the following modules, which are described in the subsequent subsections:

- p2b
- tx_dphy
- Glue module:
    - p2m_csi2_glue
    - p2m_dsi_glue

3.1.1. p2b

You must create this module to convert pixel data into byte data output according to configurations, such as TX interface, data type, number of TX lanes, and other settings. The following figure shows an example of IP interface settings in the Lattice Diamond software for the Pixel-to-Byte Submodule IP. Refer to the Pixel-to-Byte Converter IP User Guide (FPGA-IPUG-02026) for details.

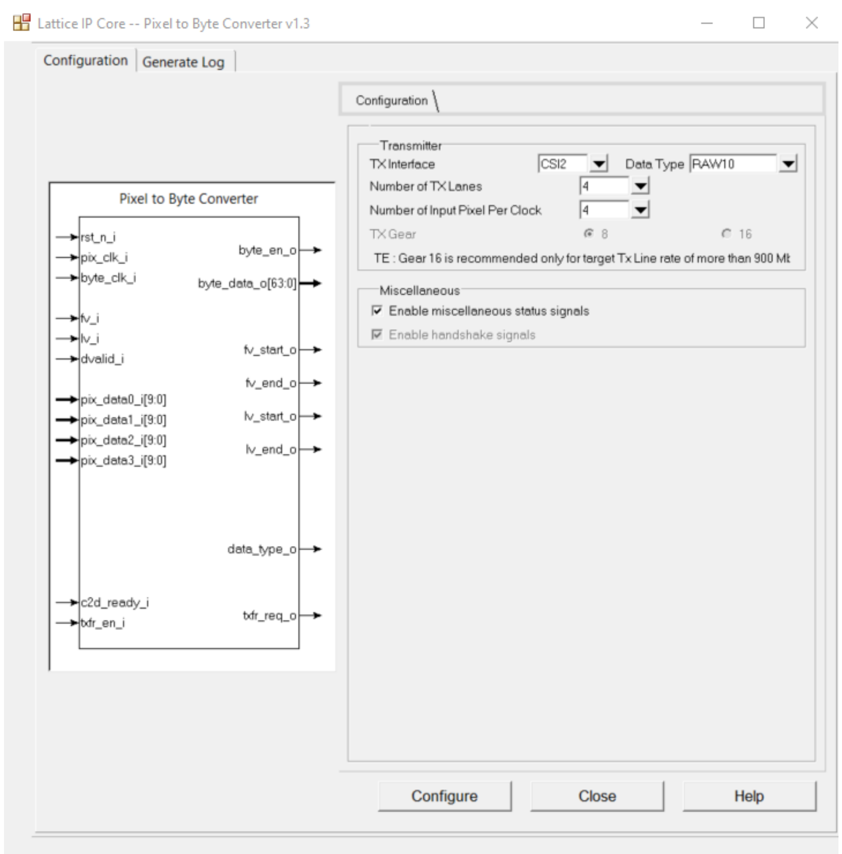

Figure 3.4. p2b IP Creation in the Lattice Diamond Software

The following guidelines and parameter settings are required for this reference design:

- TX Interface—Select DSI or CSI-2. Set the same type as TX D-PHY IP.
- Data Type—Select RGB888 or RGB666 for DSI and RGB888, RAW8, RAW10, or RAW12 for CSI-2. Other data types are not supported in this reference design.
- Number of TX Lanes—Select 1, 2, or 4. Set the same value as TX D-PHY IP.
- Number of Input Pixel Lanes—Select 1, 2, 4, 6, 8, and 10 for input pixel per clock. Number of Input Pixel Per Clock 6, 8, and 10 are only supported for CSI-2, RAW10, and RAW12.
- TX Gear—Select 8 or 16. Set according to the configuration.
- Enable miscellaneous status signals—Select checkbox to enable (checked).

The Pixel-to-Byte Converter IP converts the standard pixel data format to the D-PHY CSI-2/DSI standard based byte data stream. The .sbx file included in the project (ip/ip.sbx) can be used to reconfigure the IP per the user configuration requirements.

3.1.2. tx_dphy

You must create this module according to the channel conditions, such as number of lanes, bandwidth, and other settings. The following figure shows an example IP interface setting in the Lattice Diamond software for the CSI-2/DSI D-PHY Transmitter Submodule IP. Refer to the CSI-2/DSI D-PHY Transmitter Submodule IP User Guide (FPGA-IPUG-02024) for details.

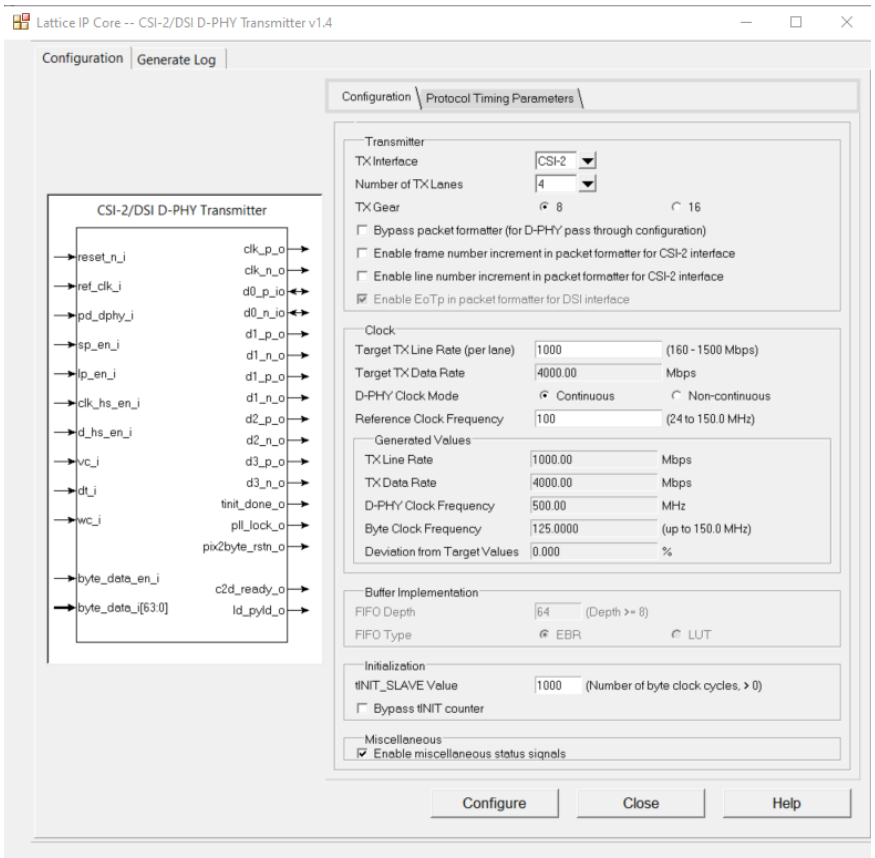

Figure 3.5. tx_dphy IP Creation in the Lattice Diamond Software

The following guidelines and parameter settings are required for this reference design:

- TX Interface Type—Select DSI or CSI-2. Set according to the required configuration.
- Number of TX Lanes—Select 1, 2, or 4. Set according to the required configuration.
- TX Gear—Select 8 or 16. Set according to configuration.
- Bypass Packet Formatter—Select disabled (unchecked).
- Enable Frame Number Increment in Packet Formatter—Select disabled (checked), only for CSI-2.
- Enable Line Number Increment in Packet Formatter—Select disabled (unchecked), only for CSI-2.
- Enable EoTp—Select checkbox to enable (checked) EoTp insertion, only for DSI. This option is not mandatory.
- Target TX Line Rate (per Lane) [160 – 1500 Mbps] —Set according to the required configuration.
- D-PHY Clock Mode—Continuous or Non-continuous. Set according to the required configuration.
- Reference Clock Frequency [24 – 150 MHz]—Set the same value as pixel clock frequency. Ensure the clock frequency able to generate TX Line Rate, TX Data Rate, and other parameters correctly under the Generated Values.
- tINIT_SLAVE Value—Use default value.

- Bypass tINIT counter—Select disabled (unchecked).
- Enable Miscellaneous Status Signals—Select checkbox to enable (checked).
- Protocol Timing Parameters tab—Default values are recommended (change timing values if required).

This module takes the byte data and outputs DSI/CSI-2 data after serialization in DSI/CSI-2 High Speed mode. The .sbx file included in the project (ip /ip.sbx) can be used to reconfigure the IP per the user configuration requirements.

Note: The Target TX Line Rate need to set according to the pixel clock frequency. Use the provided *p2m_helper_calculator.xlsx* file located within the *<design_directory>/doc* folder to determine the correct parameters value.

3.1.3. Glue module

To send the pixel data over from p2b module to the tx_dphy module, proper sequence of handshaking is required as described in the Pixel-to-Byte Converter IP User Guide (FPGA-IPUG-02026) and CSI-2/DSI D-PHY Transmitter Submodule IP User Guide (FPGA-IPUG-02024). The CrossLink Parallel-To-MIPI for reference design includes two additional modules that act as a glue logic between p2b and tx_dphy modules.

The following figure shows the simplified state machine diagrams of the hand shaking between p2b and tx_dphy modules. Both variants of the glue modules share the same initial handshaking for tx_dphy to be ready to send high-speed data packets over the MIPI D-PHY interface.

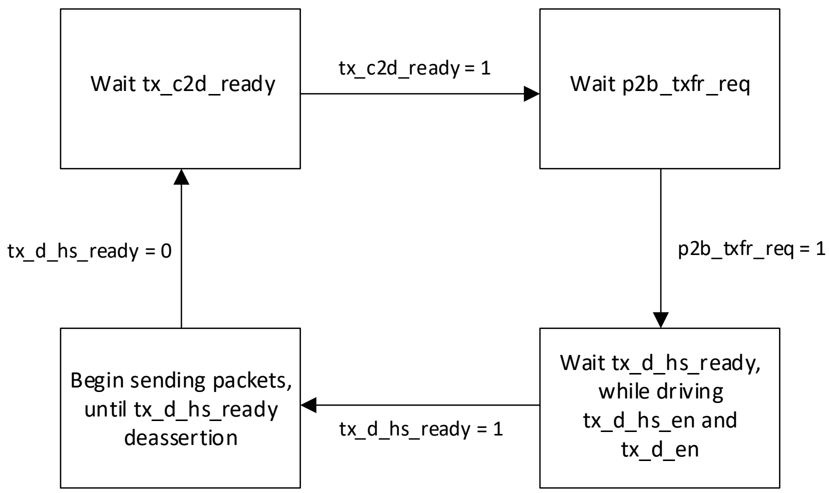

Figure 3.6. State Machine for Hand Shaking Between the p2b and tx_dphy Modules

The following figure shows the timing diagram of the initial hand shaking signals between p2b and tx_dphy modules. For each packet transaction, the initial hand shaking sequence begins with the assertion of c2d_ready signal of tx_dphy module (tx_c2d_ready_i). Note that the glue module assumes that there are sufficient blanking periods between transactions for the tx_dphy to complete each of them. The c2d_ready signal of tx_dphy needs to be asserted during the blanking period to indicate that the tx_dphy has completed sending the previous packet.

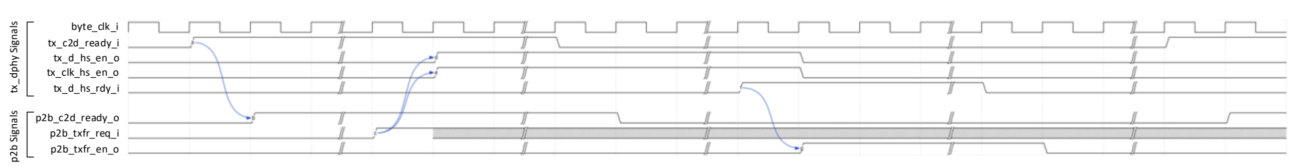

Figure 3.7. Initial Hand Shaking Timing Diagram Between the p2b and tx_dphy Modules

When the txfr_en signal of the p2b module (p2b_txfr_en_o) is asserted, the p2b module begins sending a short or long packet. The handling of the packets is slightly different between CSI-2 and DSI protocols.

3.1.3.1. p2m_csi2_glue

The p2m_csi2_glue module is the glue module that specifically handles short and long CSI-2 packets transaction between p2b and tx_dphy modules.

The following figure shows the short packet transfer timing diagram for frame valid start and frame valid end packets. Note that the design does not implement the optional line valid start or line valid end packets transfers. The short packet enable flag (tx_sp_en_o) is pulsed for one cycle after the detection of frame valid start (p2b_fv_start_i) or frame valid end (p2b_fv_end_i) from the p2b module. The header data (tx_dt_o, tx_vc_o and tx_wc_o) corresponding to the packet is sent over and remains at fix value during the transfer until the p2b_txfr_en_o de-assertion. The value of VC of tx_vc_o is configurable as a module parameter and is defaulted to 0.

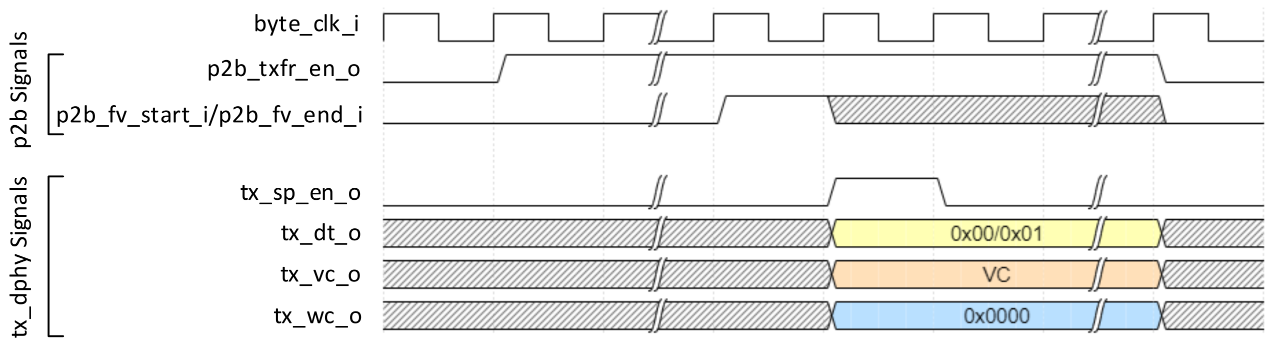

Figure 3.8. p2b to tx_dphy CSI-2 Short Packet Transfer Timing Diagram

The following figure shows the long packet transfer timing diagram of CSI-2 packets. The byte_en signal from p2b module (p2b_byte_en_i) triggers lp_en of tx dphy (tx_lp_en_o) for one clock cycle. To meet the D-PHY Tx IP requirements, the byte data and byte data enable of the tx_dphy module need to be three clock cycles away after the de-assertion of the lp_en signal. This requirement translates to the delay of the byte data from p2b to tx_dphy module by five clock cycles. The packet headers are sent over and remain at fix value throughout the transfer until the p2b_txfr_en_o de-assertion. The value of tx_dt_o DT corresponds to the data type code as defined by the CSI-2 specifications. The value of tx_wc_o WC is the number of bytes to be sent over in this transfer, which is dependent on the pixel data type and number of pixels in a line. The number of pixels in a line is fix throughout the design and is defined the synthesis directives NUM_PIXELS.

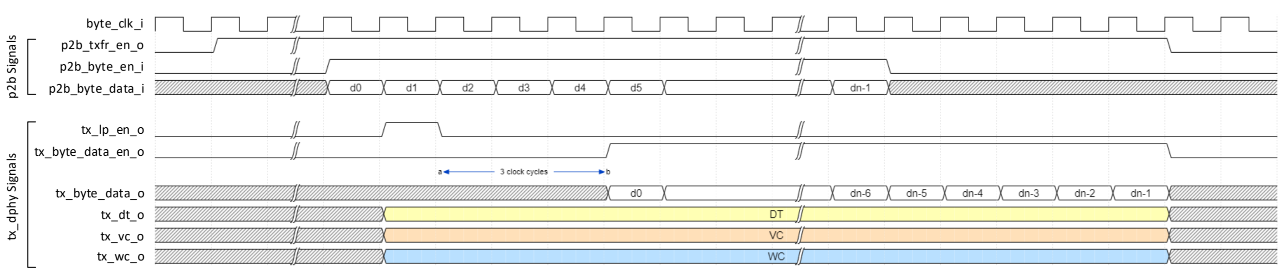

Figure 3.9. p2b to tx_dphy CSI-2 Long Packet Transfer Timing Diagram

3.1.3.2. p2m_dsi_glue

The p2m_dsi_glue module is the glue module that specifically handles short and long DSI packets transaction between p2b and tx_dphy modules.

The following figure shows the short packet transfer timing diagram for vsync and hsync start packets. Note that the design supports only non-burst with sync event mode, thus vsync and hsync end packets are not transferred to the tx_dphy module. The sync start flag of the tx_dphy (tx_vsync_start_o or tx_hsync_start_o) is pulsed for one cycle after the detection of sync start of p2b (p2b_vsync_start_i or p2b_hsync_start_i). The header data is sent over in similar manner as the p2m_csi2_glue module.

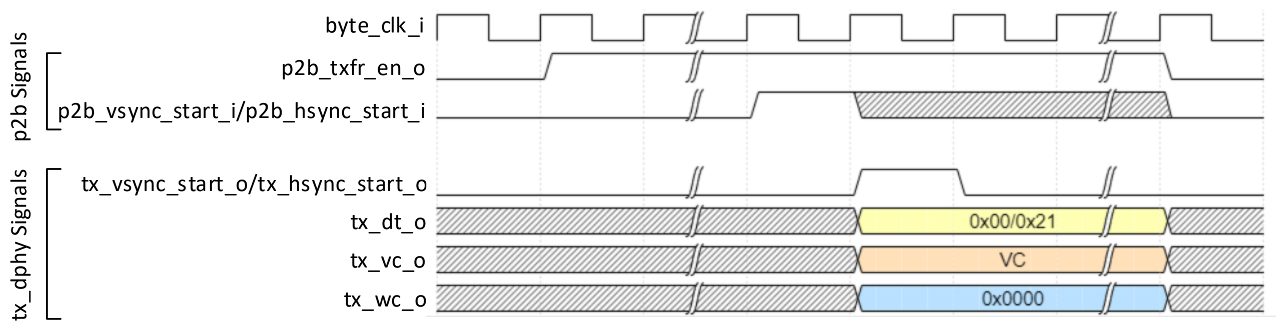

Figure 3.10. p2b to tx_dphy DSI Short Packet Transfer Timing Diagram

The following figure shows the long packet transfer timing diagram of DSI packets. Unlike the p2m_csi2_glue module, the p2m_dsi_glue module triggers the byte transfer on the next clock cycle after the detection of the byte enable of p2b (p2b_byte_data_i). The header data is sent over in similar manner as the p2m_csi2_glue module.

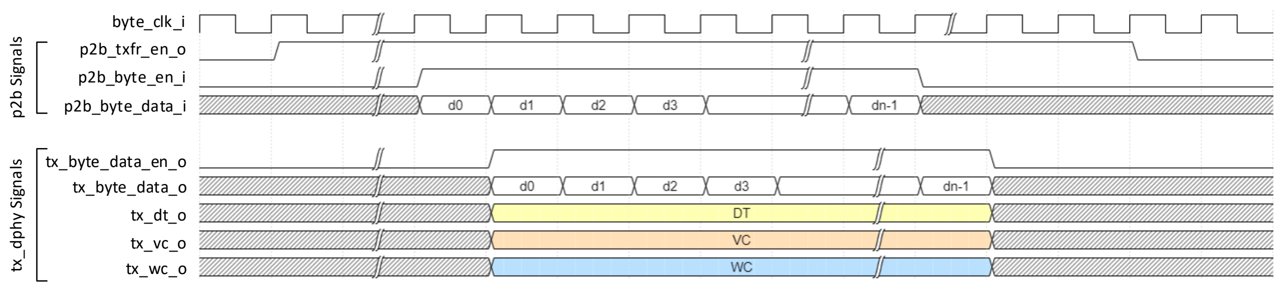

Figure 3.11. p2b to tx_dphy DSI Long Packet Transfer Timing Diagram

3.2. Clocking Scheme

This section provides the overview of the reference design clocking scheme.

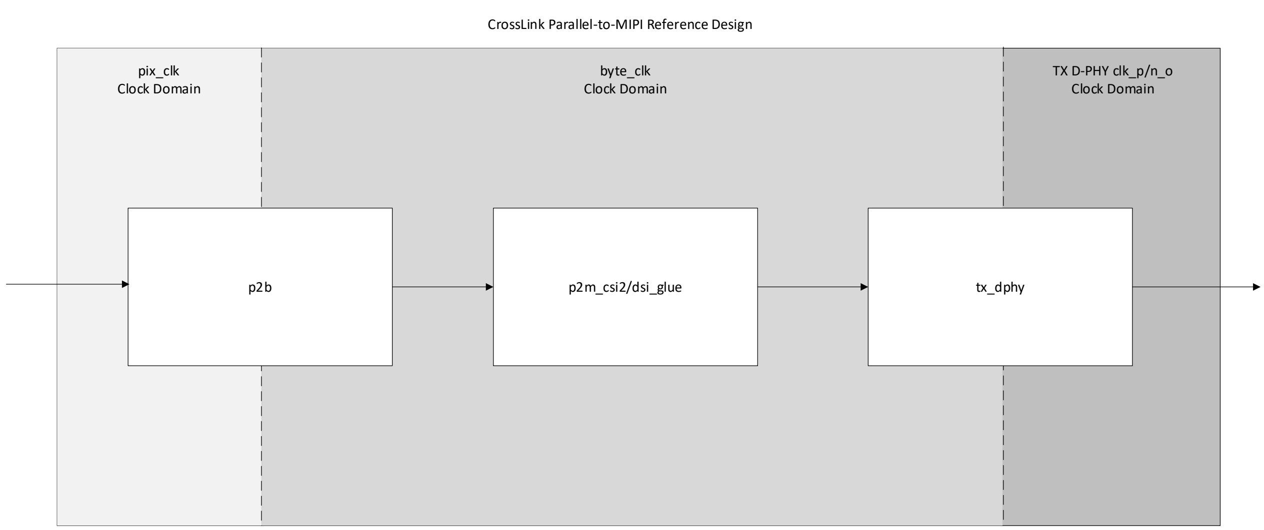

Figure 3.12. Reference Design Clock Domain Block Diagram

The figure shows the clock domains of the data path of the designs.

- Pixel clock domain (pix_clk) is used when the data is already in pixel format. Sync signals also use this domain. This clock must be supplied externally.

- TX byte clock domain (byte_clk) is used when the data is in byte format and to be consumed by tx_dphy. This clock is generated by internal PLL of hard D-PHY module.
- The TX D-PHY clock domain (TX D-PHY clk_p/n_o) is used to transmit D-PHY packets over the D-PHY channel. The clock is generated by internal PLL of hard D-PHY 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 modules, including the p2m_csi2_glue or p2m_dsi_glue module. After the system reset is released, all modules are out of reset.

This design also uses reset signal to control the power down signal for TX D-PHY. The de-assertion of reset_n_i signal asserts the tx_pd_dphy signal.

4. Reference Design Parameter Description

There are three directives files for this reference design:

- synthesis_directives.v—used for design compilation by the Lattice Diamond software and for simulation.
- simulation_directives.v—used for simulation.
- tb_param.v—used for simulation.

You can modify these directives according to your configuration. The settings in these files must match the Pixel-to-Byte and TX D-PHY IP settings created by the Lattice Diamond software for the design to work.

4.1. Synthesis Directives

The following table shows the synthesis directives that affect this reference design. These directives are used for both synthesis and simulation. Some parameter selections are restricted by other parameter settings.

Table 4.1. Parameters in synthesis_directives.v

Parameter	Default Value	Description
TX_DSI TX_CSI2	TX_CSI2	Defines the protocol to be used.
RGB888 RGB666 RAW8 RAW10 RAW12	RAW8	Defines the data types to be transmitted or received.
NUM_TX_LANE_1 NUM_TX_LANE_2 NUM_TX_LANE_4	NUM_TX_LANE_4	Number of MIPI D-PHY lanes to be used in the design.
NUM_PIX_LANE_1 NUM_PIX_LANE_2 NUM_PIX_LANE_4 NUM_PIX_LANE_6 NUM_PIX_LANE_8 NUM_PIX_LANE_10	NUM_PIX_LANE_4	Defines the number of pixel lanes or pixel per clock. Note that not all modes are available for a given combination of data types and number of lanes.
TX_GEAR_8 TX_GEAR_16	TX_GEAR_8	Defines the number of gears used in the design.
MISC_ON MISC_OFF	MISC_ON	Enables internal signals monitored by testbench. Must define only one directive.
VC	2’b00	Virtual channel ID.
NUM_PIXELS	1920	Number of active pixels per line.
CLK_MODE_HS_ONLY[1] CLK_MODE_HS_LP[1]	CLK_MODE_HS_ONLY	TX D-PHY clock mode. Must define only one directive.

Note:

1. CLK_MODE_HS_LP mode means *non-continuous clock mode* and CLK_MODE_HS_ONLY means *continuous clock mode* for the TX D-PHY.

4.2. Simulation Directives

The following table shows the simulation directives for this reference design.

Table 4.2. Parameters in simulation_directives.v

Parameter	Default Value	Description
PIX_CLK	10	Pixel clock period in ns.
NUM_FRAMES	3	Number of video frames to be transmitted.
NUM_LINES	3	Number of active lines per frame.

Parameter	Default Value	Description
HFRONT	NUM_PIXELS * 0.5	Horizontal Front Porch. Number of blanking cycles before HSYNC signal is asserted.
HPULSE	NUM_PIXELS * 0.15	Number of cycles for which HSYNC signal is asserted.
HBACK	NUM_PIXELS * 0.35	Horizontal Back Porch. Number of blanking cycles after HSYNC signal is deasserted.
VFRONT	1	Vertical Front Porch. Number of blanking lines before VSYNC signal is asserted.
VPULSE	3	Number of lines for which VSYNC signal is asserted.
VBACK	5	Vertical Back Porch. Number of blanking lines after VSYNC signal is deasserted.

4.3. Testbench parameters

The following table shows the testbench parameters for design simulation.

Table 4.3. Parameters in tb_params.v

Parameter	Default Value	Description
HEADER_CHECK	On	Enables or disables Header Data checker in simulation.
PKT_FORMAT	On	Enables or disables Packet Format checker in simulation.
EOTP_ENABLE	On	Enables or disables EOTP checker in simulation.
CRC_CHECK	On	Enables or disables CRC checker in simulation.
TIMING_PARAM_CHECK	On	Enables or disables Timing Parameter checker in simulation.

5. Signal Description

Table 5.1. Primary I/O for the p2m_top Module

Port Name	I/O	Width	Description
pix_clk_i	Input	1	Input pixel or reference clock. Period of pixel clock is defined in simulation_directives.v.
reset_n_i	Input	1	Asynchronous active low system reset.
pll_lock_o[3]	Output	1	D-PHY PLL lock signal.
fv_i[2]	Input	1	Input frame valid for parallel interface.
lv_i[2]	Input	1	Input line valid sync for parallel interface.
dvalid_i[2]	Input	1	Input data enable for parallel interface.
vsync_i[1]	Input	1	Input vertical sync for parallel interface.
hsync_i[1]	Input	1	Input horizontal sync for parallel interface.
de_i[1]	Input	1	Input data enable for parallel interface.
pixdata_i	Input	PIX_WIDTH × NUM_PIX_LANE	Input pixel data. Data Bus width depends on the data type selected and number of pixels per clock.
			RGB888: 24-bit bus width × Number of Pixel per clock
			RGB666[1]: 18-bit bus width × Number of Pixel per clock
			RAW12[2]: 12-bit bus width × Number of Pixel per clock
			RAW10[2]: 10-bit bus width × Number of Pixel per clock
			RAW8[2]: 8-bit bus width × Number of Pixel per clock
d_p_io[4]	Input/Output	NUM_TX_LANE	Positive differential TX D-PHY data lanes.
d_n_io[4]	Input/Output	NUM_TX_LANE	Negative differential TX D-PHY data lanes.
clk_p_io	Input/Output	1	Positive differential TX D-PHY clock lanes.
clk_n_io	Input/Output	1	Negative differential TX D-PHY clock lanes.

Notes:

1. Available only if data interface is DSI.
2. Available only if data interface is CSI-2.
3. Turned on if the *Enable miscellaneous status signals* attribute is selected.
4. NUM_TX_LANE = Number of TX D-PHY Lanes: 1, 2, 4 (available on the user interface).

6. Running the Reference Design

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

6.1. Compiling the Reference Design

This section provides the procedure of compiling and creating your FPGA bitstream file using the Lattice Diamond software. The full design compilation is required each time you perform IP generation, or any modifications in the design including modification in the synthesis_directives.v.

To compile your design using the Lattice Diamond software, follow these steps:

1. Open the Lattice Diamond software. The main page of the software is shown in the following figure.

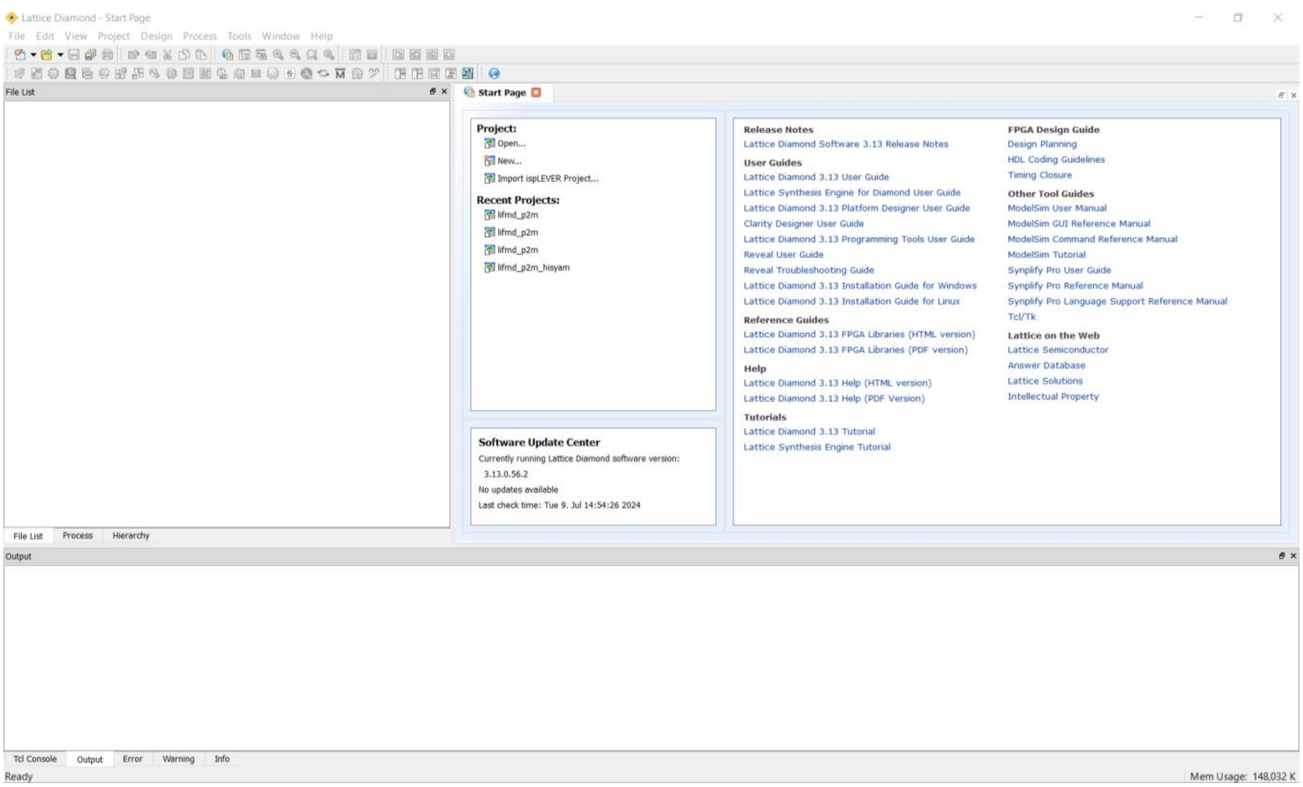

Figure 6.1. The Lattice Diamond Software

2. Click Open... and from the project database, open the Lattice Diamond software design file (.ldf) from the design folder.
   Note: You must set constraint and pin assignment to match with your hardware requirement.
3. Within the Process tab, double-click Export Files to perform full design compilation and to generate the bitstream file.

7. Simulating the Reference Design

The script file (parallel2mipi_msim.do) and testbench files are provided to run the functional simulation by ModelSim. To successfully run the simulation, you must correctly define the project directory in the parallel2mipi_msim.do file.

```
### Set Customer's project/simulation directory ###

set diamond_dir C:/lscc/diamond/3.13
set project_dir C:/lscc/diamond/3.13/workspace/Project/lifmd_p2m_RD_v1p0
set sim_dir     $project_dir/simulation
```

Figure 7.1. Project Directory Location in the parallel2mipi_msim.do File

You need to modify simulation_directives.v and tb_params.v according to your test requirement (refer to Table 4.2 and Table 4.3 for details). By executing the script in ModelSim, compilation and simulation are executed automatically.

To execute the script in Modelsim, follow these steps:

1. In the Lattice Diamond software, click the M button to open ModelSim.
2. Enter the following command in ModelSim Transcript to run the script:
   ```
   ModelSim > do simulation/parallel2mipi_msim.do
   ```
3. Simulation starts. When the test is complete, simulation stops.

7.1. Simulation Results

The parallel2mipi_tb.v testbench instantiates the top-level design module, generates the stimulus video data, and compares the expected data and the output data from the reference design, including EoT packet check, CRC check, EoTp (Long Packet and Short Packet), ECC, and timing parameters of TX D-PHY. The testbench shows the following statements while running the simulation.

```
#            0       ===============================
#            0       D-PHY Type = DSI
#            0       Data Type = RGB888
#            0       No of TX LANE = 4
#            0       No of PIX/CLK = 1
#            0       TX Gear = 8
#            0       Clock Mode = HS_ONLY
#            0       D-PHY_IP = LATTICE
#            0       ===============================
#            0       TEST START
#            0       Num of Frames : 3
#            0       Num of Lines per Frame : 3
#            0       ===============================
#          525       test_hsync_front_porch : 480
#          525       test_hsync_width : 288
#          525       test_hsync_back_porch : 672
#          525       test_h_width : 1920
#          525       test_v_height : 3
#          525       test_vsync_front_porch : 1
#          525       test_vsync_width : 3
#          525       test_vsync_back_porch : 36
#          525       test_number_of_bytes : 5760
#
#      1008359       FRAME #1 START
#      1060322       LINE #1 Transmitted
```

```
#       1088318       LINE #2 Transmitted
#       1116313       LINE #3 Transmitted
#       1148337       FRAME #1 END
#
#
#       1232324       FRAME #2 START
#       2264129       LINE #1 Transmitted
#       2292125       LINE #2 Transmitted
#       2320120       LINE #3 Transmitted
#       2352144       FRAME #2 END
#
#
#       2436131       FRAME #3 START
#       3467937       LINE #1 Transmitted
#       3495932       LINE #2 Transmitted
#       3523928       LINE #3 Transmitted
#       3555952       FRAME #3 END
```

When the simulation is finished, the following statements are displayed.

```
#       3640032       ---------------------------------------------
#       3640032       ##### DATA COMPARING IS STARTED #####
#       3640032       ---------------------------------------------
#       3640032       ***PASS : EOT PACKET CHECK***
#       3640032       ***PASS : SYNC CHECK
#       3640032       ***PASS : ECC
#       3640032       ***PASS : EoTp PACKET CHECK AFTER LP***
#       3640032       ***PASS : EoTp PACKET CHECK AFTER SP***
#       3640032       ***PASS : TIMING PARAMETERS***
#       3640032       ***PASS : CRC***
#       3640032       Test fail count : 0
#
#       3640032       -----------------------------------------------------
#       3640032       ----------------- SIMULATION PASSED -----------------
#       3640032       -----------------------------------------------------
```

The testbench generates other debug files during simulation such as, *input_data.log*, *output_data.log*, and *dphy_checker_timing.log* for debugging purpose. The *input_data.log* file stores the data transmitted by the testbench. The *output_data.log* file stores the data received to the testbench. The testbench compares both of these files. The *dphy_checker_timing.log* file stores all the timing parameters (such as LP-11, TLPX, HS-prepare, HS-0, and HS-Trail) and gives error if any timing parameter fails. The same file also saves the timing of Header Packet received and Header Packet values like DT, VC, WC, and ECC.

The following figure show the simulation waveform of the full view of three lines and three frames for the DSI: RGB888 interface.

```
Calculation for DSI – RGB888:
1920x1080p@60Hz, 4-lane, 8 Gears, 1 Pixel Lane, Continuous Mode

Total Horizontal Samples = 2200
Total Vertical Lines = 1125
**Refer to MIPI D-PHY Bandwidth Matrix and Implementation Table 2.1. Common Video Format.

Pixel Clock Frequency
```

```
PCF = 2200 x 1125 x 60 = 148.5 MHz
**Input this frequency at reference clock in your tx_dphy.

Bandwidth (Total Data Rate)
B = 148.5 MHz x 24-bit = 3.564 Gbps
**RGB888 uses 24 bits.

Line Rate (Data Rate per Lane)
LR = 3.564 Gbps/4-lane = 891 Mbps
**Input this tx line rate at tx_dphy. Maximum TX bandwidth is 1.5 Gbps/lane
```

Note: For DSI Simulation, Eopt enable (tx_dphy) is needed to be enabled.

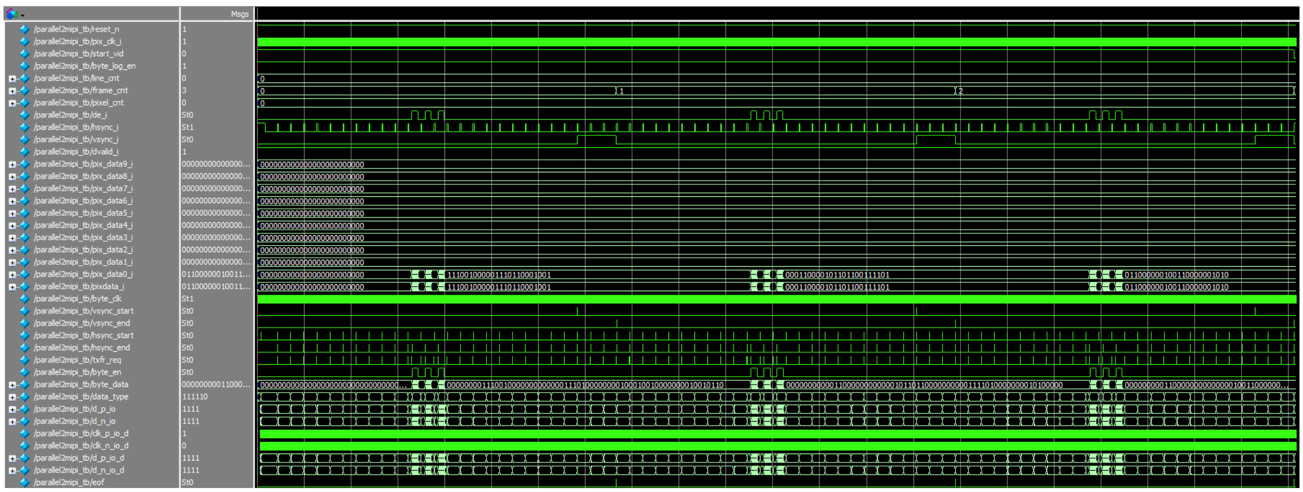

Figure 7.2. Simulation Waveform for DSI: RGB888

The following figure shows the simulation waveform of the full view of three lines and three frames for the CSI-2: RAW10 interface.

```
Calculation for CSI-2 – RAW10:
1920x1080p@60Hz, 4-lane, 8 Gears, 1 Pixel Lane, Continuous Mode

Total Horizontal Samples = 2200
Total Vertical Lines = 1125
**Refer to MIPI D-PHY Bandwidth Matrix and Implementation Table 2.1. Common Video Format.

Pixel Clock Frequency
PCF = 2200 x 1125 x 60 = 148.5 MHz
**Input this frequency at reference clock in your tx_dphy.

Bandwidth (Total Data Rate)
B = 148.5 MHz x 10-bit = 1.485 Gbps
**RAW10 uses 10 bits.

Line Rate (Data Rate per Lane)
LR = 1.485 Gbps/4-lane = 371.25 Mbps
**Input this tx line rate at tx_dphy. Maximum TX bandwidth is 1.5 Gbps/lane.
```

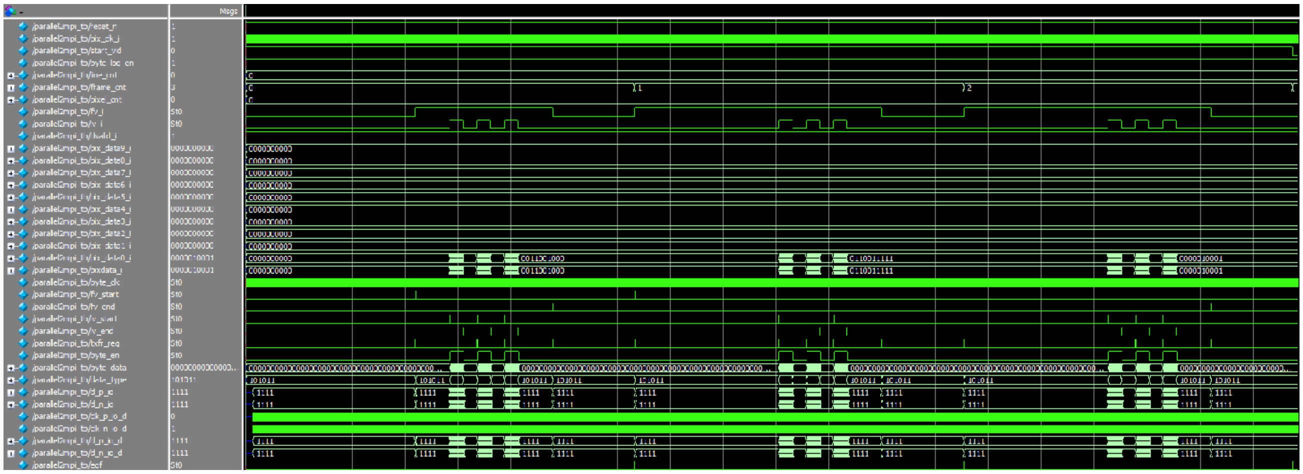

Figure 7.3. Simulation Waveform for CSI-2: RAW10

The following figure shows the simulation waveform of the full view of three lines and three frames for the CSI-2: RAW12 interface. The waveform shows all the top level I/O and a few other signals.

```
Calculation for CSI-2 – RAW12:
1920x1080p@60Hz, 4-lane, 8 Gears, 1 Pixel Lane, Continuous Mode

Total Horizontal Samples = 2200
Total Vertical Lines = 1125
**Refer to MIPI D-PHY Bandwidth Matrix and Implementation Table 2.1. Common Video Format.

Pixel Clock Frequency
PCF = 2200 x 1125 x 60 = 148.5 MHz
**Input this frequency at reference clock in your tx_dphy.

Bandwidth (Total Data Rate)
B = 148.5 MHz x 12-bit = 1.782 Gbps
**RAW12 uses 12 bits.

Line Rate (Data Rate per Lane)
LR = 1.782 Gbps/4-lane = 445.5 Mbps
**Input this tx line rate at tx_dphy. Maximum TX bandwidth is 1.5 Gbps/lane.
```

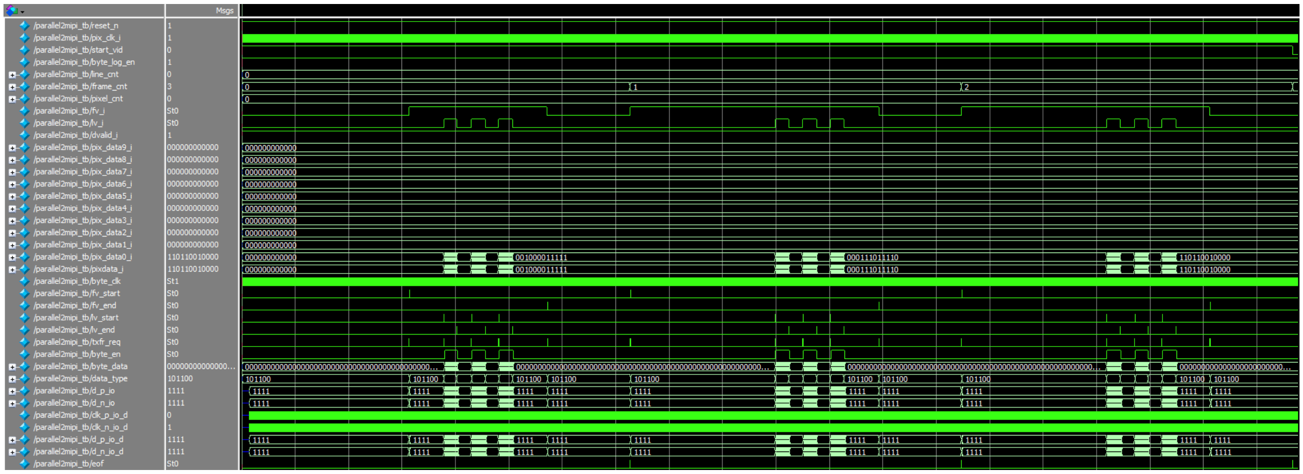

Figure 7.4. Simulation Waveform for CSI-2: RAW12

8. Known Limitations

The following are the limitations of this reference design:

- Only the following data types are supported for the MIPI DSI interface: RGB888 and RGB666
- Only the following data types are supported for the MIPI CSI-2 interface: RGB888, RAW8, RAW10, and RAW12
- For DSI mode, only non-burst sync event mode is supported.
- For CSI-2 mode, only frame synchronization (Frame Start and Frame End) packets are implemented.
- For CSI-2 mode, frame number is inoperative, for example, frame number is set to 0 when sending out the respective short packets.

9. Resource Utilization

Resource utilization depends on the configuration used. The values in following table are a reference and the actual usage varies.

Table 9.1. Resource Utilization Examples for the LIF-MD6000 Device

Configuration	LUT	FF	EBR	I/O
4-lane, Gear 16, DSI, RGB888, 4 pixel/clock	1,005	621	16	3
1-lane, Gear 8, DSI, RGB888, 1 pixel/clock	323	305	3	3
4-lane, Gear 16, CSI-2, RAW12, 1 pixel/clock	1,194	816	4	17
1-lane, Gear 8, CSI-2, RAW8, 1 pixel/clock	430	435	1	13

10. Debugging

This section lists the suggested troubleshooting methods and tools that you can use.

10.1. Debug Methods

10.1.1. Dphy_checker_timing.log

When there is a simulation failure caused by timing parameters, ModelSim Transcript shows the error messages in the following example.

```
#       3640032       ---------------------------------------------
#       3640032       ##### DATA COMPARING IS STARTED #####
#       3640032       ---------------------------------------------
#       3640032       ***PASS : EOT PACKET CHECK***
#       3640032       ***PASS : SYNC CHECK
#       3640032       ***PASS : ECC
#       3640032       ***PASS : EoTp PACKET CHECK AFTER LP***
#       3640032       ***PASS : EoTp PACKET CHECK AFTER SP***
#       3640032       ***FAIL : TIMING PARAMETERS***
#       3640032       ***PASS : CRC***
#       3640032       Test fail count : 10
#
#       3640032       -----------------------------------------------------
#       3640032       ----------------- SIMULATION FAILED! ----------------
#       3640032       -----------------------------------------------------
```

To debug the failure, open the *dphy_checker_timing.log* file located in the simulation directory and read the error messages. The log provides recommendations on the correct timing parameters value for the current configuration, and you need to make the changes accordingly in the DPHY TX IP timing parameter.

10.2. Debug Tools

10.2.1. Configuration Helper Calculator

The reference design includes the *p2m_helper_calculator.xlsx* file, located within the *<design_directory>/doc* folder. This calculator assists in calculating the clocking requirement such as the DPHY clock, pixel clock, and byte clock for your configuration.

10.2.2. 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 ModelSim™.
- 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 details on how to use the Reveal Analyzer.

10.2.3. ModelSim

The Mentor® ModelSim® tool is an OEM simulation tool that is closely linked to the Diamond software environment. The tool can be used to perform functional verification of your design and IP. A proper testbench needs to be written to provide input stimulus to the Device Under Test (DUT) and observe the output signals via the ModelSim Waveform Viewer to verify the correctness of the IP or design. The reference design package includes the reference design testbench and other simulation files. You can use this package to verify the behavior of the design and as a reference during your debug activity. To run the simulation in ModelSim, refer to the Simulating the Reference Design section.

References

- Pixel-to-Byte Converter IP User Guide (FPGA-IPUG-02026)
- CSI-2/DSI D-PHY Transmitter Submodule IP User Guide (FPGA-IPUG-02024)
- Reveal User Guide
- CrossLink web page
- Parallel to MIPI CSI-2 / DSI Display Interface Bridge Reference Design web page
- Pixel to Byte Converter IP Core web page
- CSI-2/DSI D-PHY Transmitter IP Core web page
- Lattice Diamond 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, please refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.0, August 2024

Section	Change Summary
All	Production release.

www.latticesemi.com