CSI-2/DSI D-PHY Transmitter Submodule IP

User Guide

FPGA-IPUG-02024-1.6

April 2021

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

Contents

Acronyms in This Document 5
1. Introduction 6
1.1. Quick Facts 6
1.2. Features 6
1.3. Conventions 7
1.3.1. Nomenclature 7
1.3.2. Data Ordering and Data Types 7
1.3.3. Signal Names 7
2. Functional Descriptions 8
2.1. Interface and Timing Diagram 10
2.1.1. Input Interface 10
2.1.2. Output Interface 12
2.2. Clock, Reset and Initialization 13
2.2.1. Reset and Initialization 13
2.2.2. Clock Domains and Clock Domain Crossing 13
2.3. Design and Module Description 15
3. Parameter Settings 16
4. IP Generation and Evaluation 18
4.1. Licensing the IP 18
4.2. Getting Started 18
4.3. Generating IP in Clarity Designer 19
4.4. Generated IP Directory Structure and Files 23
4.5. Running Functional Simulation 25
4.6. Simulation Strategies 26
4.7. Simulation Environment 26
4.8. Instantiating the IP 27
4.9. Synthesizing and Implementing the IP 27
4.10. Hardware Evaluation 28
4.10.1. Enabling Hardware Evaluation in Diamond 28
4.11. Updating/Regenerating the IP 28
4.11.1. Regenerating an IP in Clarity Designer 28
References 29
Technical Support Assistance 29
Appendix A. Resource Utilization 30
Appendix B. What is Not Supported 32
Appendix C. PLL Programming 33
Revision History 36

Figures

Figure 2.1. CSI-2/DSI D-PHY Transmitter Submodule Top-level Block Diagram 8
Figure 2.2. D-PHY Tx Input Bus for Short Packet Transmission....10
Figure 2.3. D-PHY Tx Input Bus for Long Packet Transmission in CSI-2 Interface 11
Figure 2.4. D-PHY Tx Input Bus for Long Packet Transmission in DSI Interface 11
Figure 2.5. D-PHY Tx Input Bus for Long Packet Transmission in DSI/CSI-2 Interface without pkt_formatter 11
Figure 2.6. High-Speed Data Transmission 12
Figure 2.7. Clock Domain Crossing Block Diagram....13
Figure 2.8. PLL Block Diagram....14
Figure 2.9. D-PHY Transmitter Submodule Block Diagram 15
Figure 4.1. Clarity Designer Window 18
Figure 4.2. Starting Clarity Designer from Diamond Design Environment 19
Figure 4.3. Configuring D-PHY Transmitter IP in Clarity Designer....20
Figure 4.4. Configuration Tab in IP User Interface....21
Figure 4.5. Protocol Timing Parameter Tab in IP User Interface 22
Figure 4.6. D-PHY Transmitter IP Directory Structure 23
Figure 4.7. Simulation Environment Block Diagram 26
Figure 4.8 D-PHY Tx configured as CSI-2 Transmitter....26
Figure 4.9. D-PHY Tx configured as DSI Transmitter....27
Figure 4.10. IP Regeneration in Clarity Designer 28

Tables

Table 1.1. CSI-2/DSI D-PHY Transmitter Submodule IP Quick Facts 6
Table 1.2. CSI-2/DSI D-PHY Transmitter Submodule IP Features Summary 6
Table 2.1. D-PHY Transmitter Submodule IP Pin Function Description 8
Table 2.2. Byte Data Bus Assignments....12
Table 2.3. Clock Domain Crossing....13
Table 2.4. PLL Operating Frequencies....14
Table 3.1. CSI-2/DSI D-PHY Tx Submodule IP Parameter Settings in User Interface 16
Table 4.1. Files Generated in Clarity Designer....23
Table 4.2. Testbench Compiler Directives 25
Table A.1. Resource Utilization[1, 3] 30
Table C.1. 5-Bit Input Divider....33
Table C.2. 8-Bit Feedback Divider 34
Table C.3. 2-Bit Output Divider....35

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
CM	Coded PLL M parameter
CN	Coded PLL N parameter
CO	Coded PLL O parameter
CSI-2	Camera Serial Interface-2
DCS	Display Command Set
DSI	Display Serial Interface
EBR	Embedded Block RAM
EoTp	End of Transmit Packet
FIFO	First In First Out buffer
HS	High Speed
IP	Intellectual Property
LP	Low Power
LUT	Lookup Table
M	PLL feedback multiplier
MIPI	Mobile Industry Processor Interface
N	PLL input clock divider
O	PLL VCO output frequency divider
PLL	Phase-Locked Loop / Phase Lock Loop
VCO	Voltage Controlled Oscillator

1. Introduction

The Lattice Semiconductor CSI-2/DSI D-PHY Transmitter Submodule IP converts 64-bit data to either DSI or CSI-2 data for Lattice Semiconductor CrossLink™ and CrossLinkPlus™ devices. This is useful for wearable, tablet, human-machine interfacing, medical equipment, and other applications.

Mobile Industry Processor Interface (MIPI®) D-PHY has become the industry's primary high-speed PHY solution for camera and display interconnection in mobile devices. It is typically used in conjunction with MIPI Camera Serial Interface-2 (CSI-2) and MIPI Display Serial Interface (DSI) protocol specifications. It meets requirements of low-power, low-noise generation, and high-noise immunity that mobile phone designs demand.

This user guide is for CSI-2/DSI D-PHY Transmitter Submodule IP design version 1.4.

1.1. Quick Facts

Table 1.1 provides quick facts about the CSI-2/DSI D-PHY Transmitter Submodule IP for CrossLink and CrossLinkPlus devices.

Table 1.1. CSI-2/DSI D-PHY Transmitter Submodule IP Quick Facts

		D-PHY Transmitter Submodule IP Configuration	
		CSI-2 4-Lane	**DSI 4-Lane**
IP Requirements	FPGA Families Supported	CrossLink/CrossLinkPlus	
Resource Utilization	Targeted Device	LIF-MD6000-6MG81I	
	Data Path Width	64-bit	64-bit
	LUTs*	1496	5548
	sysMEM™ EBRs	0	0
	Registers	517	656
	MIPI D-PHY	1	1
Design Tool Support	Lattice Implementation	Lattice Diamond® 3.11 SP1	
	Synthesis	Lattice Synthesis Engine	
		Synplify Pro® N-2018.03L-SP1-1	
	Simulation	Aldec® Active HDL™ 10.5 Lattice Edition	

*__Note__: DSI packet formatter includes a FIFO while CSI-2 packet formatter does not. In addition, 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*.

1.2. Features

The key features of the CSI-2/DSI D-PHY Transmitter Submodule IP include:

- Compliant with MIPI DSI v1.1, MIPI CSI-2 v1.1, and MIPI D-PHY v1.1 Specifications
- Supports MIPI DSI and MIPI CSI-2 interfacing up to 6 Gb/s
- Supports 1, 2, 3, or 4 MIPI D-PHY data lanes
- Supports Burst Mode, Non-Burst Mode with Sync Pulses and Non-Burst Mode with Sync Events for transmission of DSI packets
- Supports low-power (LP) mode during vertical and horizontal blanking

Table 1.2. CSI-2/DSI D-PHY Transmitter Submodule IP Features Summary

IP Configuration	Options
Number of Tx lanes	1, 2, 3, 4
Gearing	8, 16
MIPI Interface	DSI, CSI-2

1.3. Conventions

1.3.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL. This includes radix indications and logical operators.

1.3.2. Data Ordering and Data Types

The most significant bit within the pixel data is the highest index.

1.3.3. Signal Names

Signal names that end with:

- *_n* are active low
- *_i* are input signals
- *_o* are output signals
- *_io* are bidirectional signals

2. Functional Descriptions

The CSI-2/DSI D-PHY Transmitter Submodule IP converts 64-bit data to either DSI or CSI-2 byte packets. The input interface consists of a 64-bit data arranged in CSI-2 or DSI data type format, a byte data enable, frame/vsync start signal, line/hsync start signal, or short/long packet enable, virtual channel, data type, EoTP, word count, and byte clock.

The output interface consists of serialized HS (High-speed) data packets following the MIPI CSI-2 or DSI specifications. It has a maximum of five lanes per channel (four data lanes and one clock lane only).

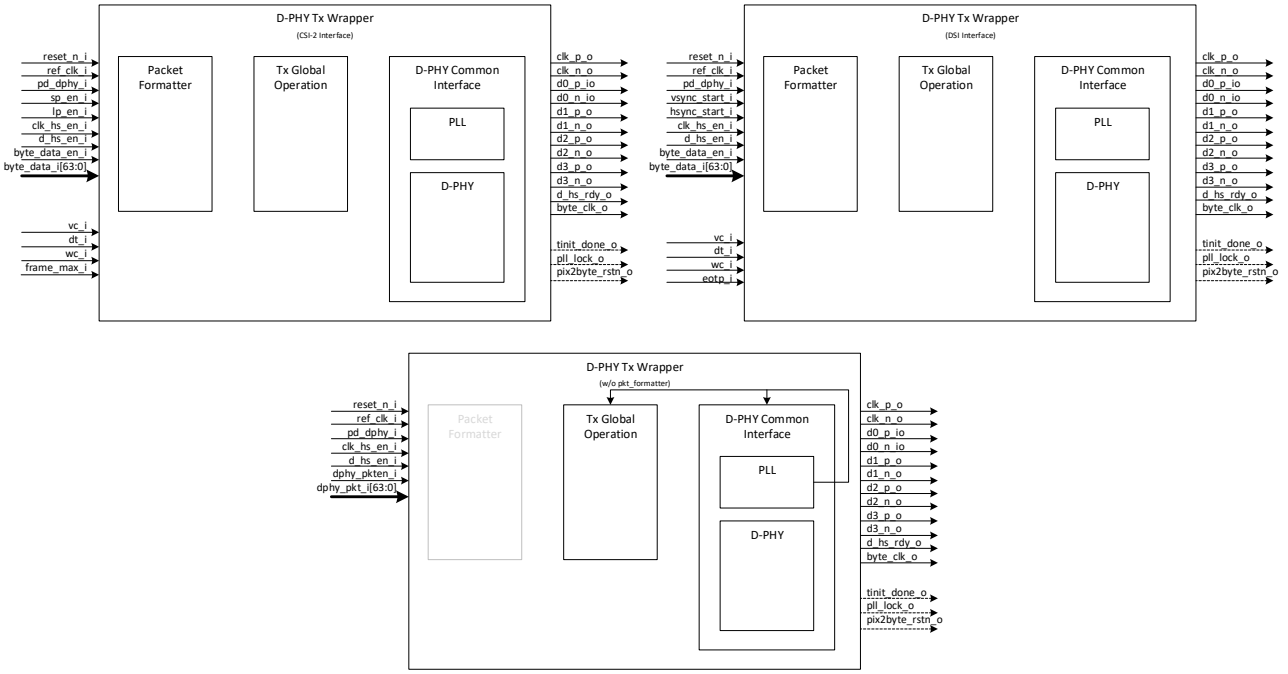

Figure 2.1. CSI-2/DSI D-PHY Transmitter Submodule Top-level Block Diagram

Table 2.1. D-PHY Transmitter Submodule IP Pin Function Description

Pin Name	Direction	Function Description
For DSI Interface		
vsync_start_i[1]	I	Start of vertical sync (active high pulse)
hsync_start_i[1]	I	Start of horizontal sync (active high pulse)
For CSI-2 Interface with Packet Formatter		
sp_en_i[2]	I	Short packet enable; this includes line start/end and frame start/end packets
frame_max_i [7:0]	I	Sets maximum frame count when the frame number count is enabled from the user interface
lp_en_i[2]	I	Long packet enable; active high pulse to request long packet creation and transmission
ld_pyld_o	O	For CSI-2, this indicates that the requestor can send the valid payload data after a long packet transmission request
Byte Data Input without Packet Formatter		
byte_data_i [63:0] [3]	I	Byte data
byte_data_en_i[3]	I	Active high byte data enable
D-PHY Packet Input		
dphy_pkt_i [63:0] [4]	I	D-PHY packet data
dphy_pkten_i[4]	I	Active high D-PHY packet enable
Common Interface		
reset_n_i	I	Asynchronous active low system reset. 0 – System is on reset
ref_clk_i	I	Reference clock for D-PHY PLL used to generate D-PHY serial clock and byte clock.

Pin Name	Direction	Function Description
pd_dphy_i	I	Power down input signal for D-PHY. When high, all D-PHY blocks are powered down. Tie low if not used.
vc_i [1:0][3]	I	Virtual channel ID.
dt_i [5:0][3]	I	Data type
wc_i [15:0][3]	I	Word Count field. This corresponds to header word1 and 2 in the header packet.
eotp_i[1]	I	End of Transmission packet enable
clk_hs_en_i[5]	I	Active high pulse going to the Tx Global operation to start LP to HS transition of clock lane
d_hs_en_i	I	Active high pulse going to the Tx Global operation to start LP to HS transition of data lane
d_hs_rdy_o	O	Active high signal that indicates the data lanes are already in high-speed mode in response to an earlier transmit request. The requestor needs to check this signal before sending a valid long or short packet.
c2d_rdy_o	O	In the case of non-continuous clock mode, this active high signal indicates the clock is in LP-11 state and is ready to accept new transmit request. For continuous clock mode, this indicates the data lanes are in LP-11 state and ready to accept new requests. The requestor needs to check this ready signal before sending a d_hs_en_i request.
byte_clk_o	O	Byte clock generated by D-PHY PLL
d0_p_io	I/O	D-PHY data lane 0 Positive Data
d0_n_io	I/O	D-PHY data lane 0 Negative Data
d1_p_o	O	D-PHY data lane 1 Positive Data
d1_n_o	O	D-PHY data lane 1 Negative Data
d2_p_o	O	D-PHY data lane 2 Positive Data
d2_n_o	O	D-PHY data lane 2 Negative Data
d3_p_o	O	D-PHY data lane 3 Positive Data
d3_n_o	O	D-PHY data lane 3 Negative Data
clk_p_o	O	D-PHY clock lane Positive end
clk_n_o	O	D-PHY clock lane Negative end
Miscellaneous		
tinit_done_o[6]	O	tINIT done signal generated from IP.
pll_lock_o[6]	O	D-PHY PLL lock signal.
pix2byte_rstn_o[6]	O	Active low done signal. Indicates the end of a long packet transmission. This signal may be used to reset the FIFOs of the requestor module.

Notes:

1. Enabled only for DSI interface (TX_DSI is defined).
2. Enabled only for CSI-2 interface (TX_CSI2 is defined).
   - lp_en_i is equivalent to the start of active line (long packet).
   - sp_en_i is equivalent to the start and end of frame or line (short packet).
3. Enabled when you need the packet formatter, and disabled when packet formatter is not needed.
4. Enabled when you do not need the packet formatter.
5. Enabled for non-continuous clock mode.
6. Can be turned-on when MISC_ON is defined.

2.1. Interface and Timing Diagram

2.1.1. Input Interface

2.1.1.1. Short Packet Transmission (Packet Formatter Enabled)

Figure 2.2 shows the timing relationship between sp_en_i (frame and line short packets) for CSI-2 interface, hsync/vsync start for DSI interface, and d_hs_en_i/clk_hs_en_i and d_hs_rdy_o signals for short packet transmission. Short packet transmission is as follows:

1. Requestor checks the c2d_ready_o signal and asserts the d_hs_en_i when it wants to send a request for HS transmission. The clk_hs_en_i is not enabled for continuous (HS_ONLY) clock mode. For non-continuous (HS_LP) clock mode, clk_hs_en_i can be asserted earlier to enable the module to start performing the clock lane LP to HS sequence, or it can be asserted with the d_hs_en_i. Enabling the clock lane to go to high-speed mode earlier reduces the transmit latency.
2. Wait for d_hs_rdy_o to assert. This indicates the data lane has completed the LP to HS transition and is now transmitting HS-ZERO bits.
3. For CSI-2 interface, the requestor sends one pulse of sp_en_i together with the valid packet header fields, for every short packet transmission requests.

   For DSI, send one pulse of vsync_start_i or hsync_start_i. Modify the data type dt_i to 0x01, 0x11, 0x21or 0x31 when transmitting V Sync Start, V Sync End, H Sync Start or H Sync End packets respectively.

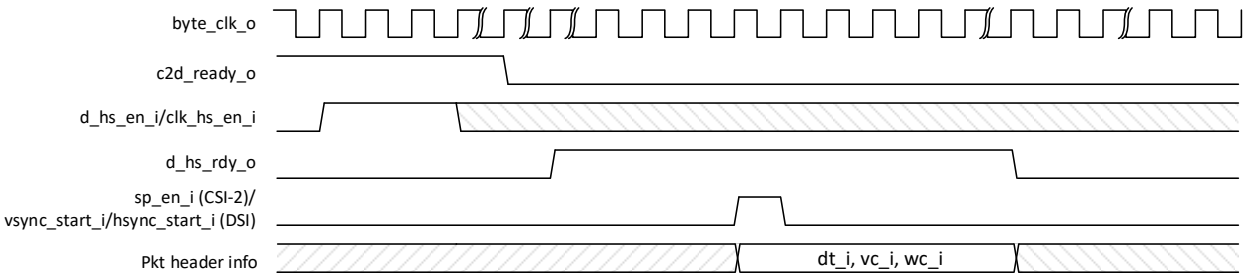

Figure 2.2. D-PHY Tx Input Bus for Short Packet Transmission

2.1.1.2. Long Packet Transmission (Packet Formatter Enabled)

Figure 2.3 and Figure 2.4 show the timing relationship between the lp_en_i (long packet enable), byte_data_en_i and byte_data. Long packet transmission is as follows:

1. Requestor checks the c2d_ready_o signal and asserts d_hs_en_i when it wants to send a request for HS transmission. The clk_hs_en_i is not enabled for continuous (HS_ONLY) clock mode. For non-continuous (HS_LP) clock mode, clk_hs_en_i needs to be asserted with the d_hs_en_i.
2. Wait for d_hs_rdy_o to assert. This indicates the data lane has completed the LP to HS transition and is now transmitting HS-ZERO bits.
3. For CSI-2 interface, send one pulse of lp_en_i together with the valid packet header fields. The signal ld_pyld_o asserts the next cycle after reception of the lp_en_i.

   For DSI, there is no lp_en_i or ld_pyld_o. Instead, the byte_data_en_i needs to be asserted one cycle earlier than the valid data.
4. For DSI, there is no lp_en_i or ld_pyld_o. Send byte_data_i together with the byte_data_en_i and the dt_i[5:0]. For CSI-2, the requestor must send the byte_data_i and byte_data_en_i 2 cycles after ld_pyld_o assertion (or 3 cycles after lp_en_i) to let the IP have enough time to create the header packets.

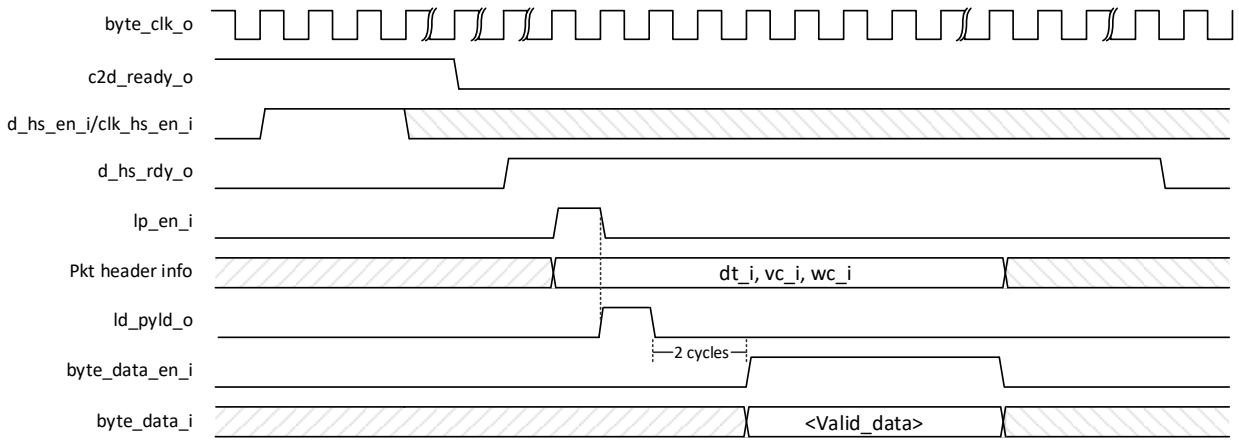

Figure 2.3. D-PHY Tx Input Bus for Long Packet Transmission in CSI-2 Interface

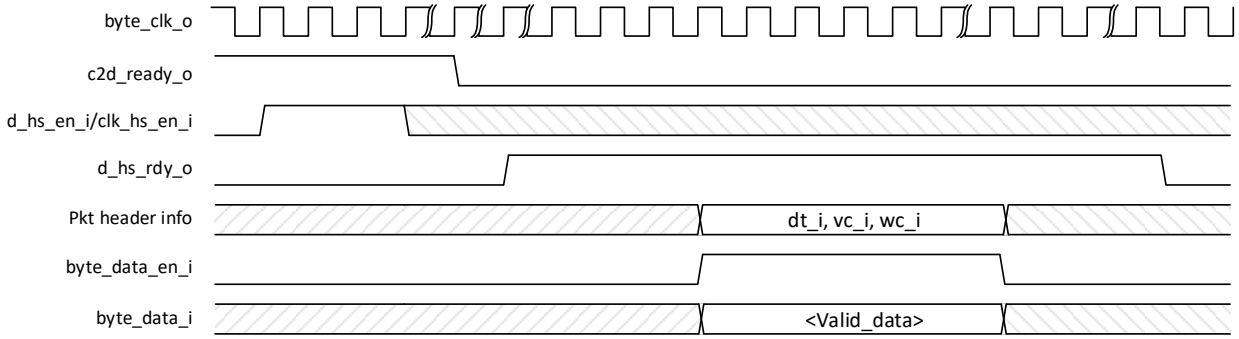

Figure 2.4. D-PHY Tx Input Bus for Long Packet Transmission in DSI Interface

2.1.1.3. Packet Transmission when Packet Formatter is Disabled

Figure 2.5 shows the timing diagram when the packet formatter module is disabled. The dphy_pkt_i should already include the sync code, packet header, payload and trail bits. After assertion of d_hs_en_i/clk_hs_en_i, wait for d_hs_rdy_o to assert and send the dphy_pkt_i with the dphy_pkten_i.

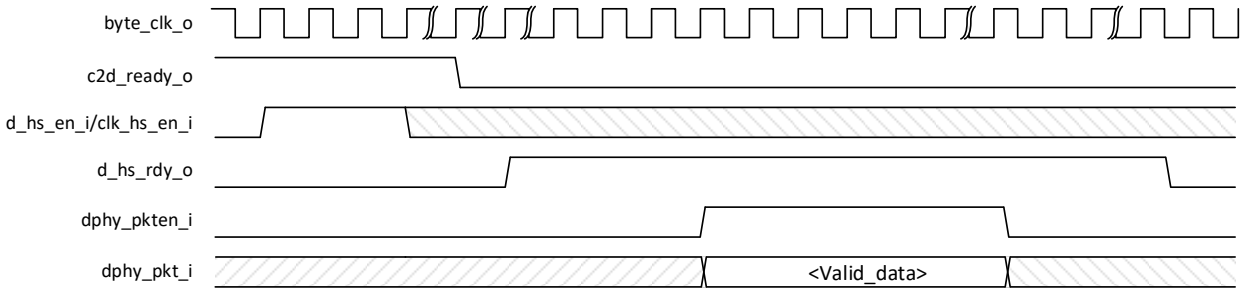

Figure 2.5. D-PHY Tx Input Bus for Long Packet Transmission in DSI/CSI-2 Interface without pkt_formatter

Table 2.2 shows the byte arrangement of the input byte_data_i in relation to how it will appear on the D-PHY data lanes.

Table 2.2. Byte Data Bus Assignments

Tx D-PHY Lanes	byte_data_i	4-Lane		2-Lane		1-Lane	
		Gear 16	Gear 8	Gear 16	Gear 8	Gear 16	Gear 8
Lane 0	[7:0]	Byte 1	Byte 1	Byte 1	Byte 1	Byte 1	Byte 1
	[15:8]	Byte 5	—	Byte 3	—	Byte 2	—
Lane 1	[23:16]	Byte 2	Byte 2	Byte 2	Byte 2	—	—
	[31:24]	Byte 6	—	Byte 4	—	—	—
Lane 2	[39:32]	Byte 3	Byte 3	—	—	—	—
	[47:40]	Byte 7	—	—	—	—	—
Lane 3	[55:48]	Byte 4	Byte 4	—	—	—	—
	[63:56]	Byte 8	—	—	—	—	—

2.1.2. Output Interface

Figure 2.6 shows that prior to the HS mode data transfer, all clock and data lanes are in the LP11 state (1.2 V on the P channel and 1.2 V on the N channel). The clock lane then goes to the LP01 state (0 V on the P channel and 1.2 V on the N channel) followed by the LP00 state (0 V on the P channel and 0 V on the N channel). After that, the clock lane goes into HS mode with SLVS200 signaling (Vcm=200 mV, Vdiff=±100 mV), and holds an HS0 state (differential 0 state of P channel = 100 mV and N channel = 300 mV when termination of the receiver is turned on) according to the MIPI D-PHY Specifications. The clock starts shortly after. When the HS clock runs, the data lanes follow a similar procedure going from LP11 to LP01, LP00, and HS0 states. The Start-of-Transmit sync pattern is driven on the line followed by the packet header and data payload. At the end of the transfer, the data lanes first go back into LP mode by going to LP00 then LP11 states. The clock lane follows shortly after.

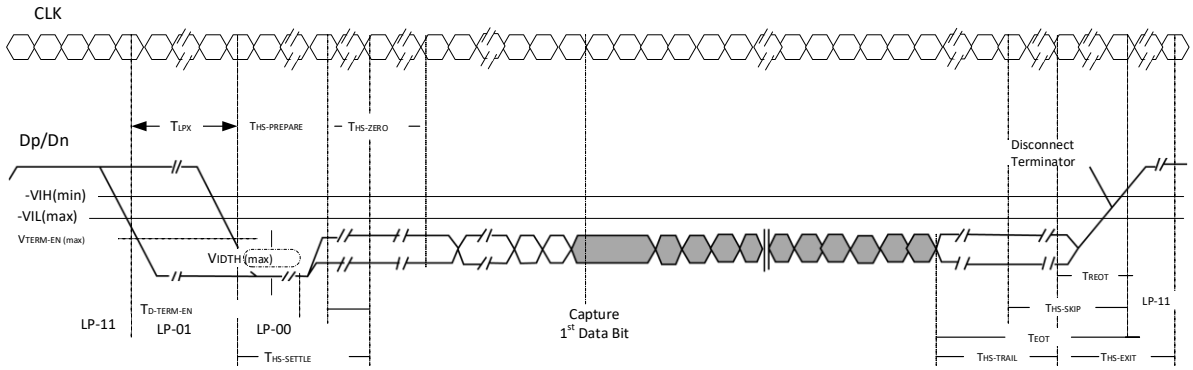

Figure 2.6. High-Speed Data Transmission

2.2. Clock, Reset and Initialization

2.2.1. Reset and Initialization

After power-up, the transmitting D-PHY is required to drive a Stop State (LP-11) for a period longer than tINIT. The D-PHY forces the lane module into transmit mode and generates the Stop State after system reset.

The Slave PHY is initialized when the Master PHY drives a Stop State (LP-11). The first Stop State that is longer than the specified tINIT is called the Initialization period. tINIT is estimated to be minimum 100 µs.

An asynchronous reset pin (active high) is used for resetting the entire FPGA. Internal reset logic is implemented to guarantee synchronous de-assertion throughout different clock domains for both hard and soft IPs. A powerdown signal pd_dphy_i also resets the hardened D-PHY block. This block must remain powered down until the internal PLL has locked. This requirement is already handled by the CSI-2/DSI D-PHY Transmitter Submodule IP. Unless specified by MIPI IP or Soft IP requirement, no special reset sequence is needed for CrossLink and CrossLinkPlus devices. However, there is a wait time requirement before the Application Processor can send valid data to the bridge. When set in DSI, until the DCS commands are sent to the display, valid data from the Application Processor might be lost. Likewise, for CSI-2, until PLL is locked, valid data from the Application Processor might be lost.

2.2.2. Clock Domains and Clock Domain Crossing

The D-PHY PLL generates the bit clock from the input reference clock. Byte clock is then derived from this bit clock through a clock generator inside the D-PHY block. Both the reference clock and byte clock are free-running.

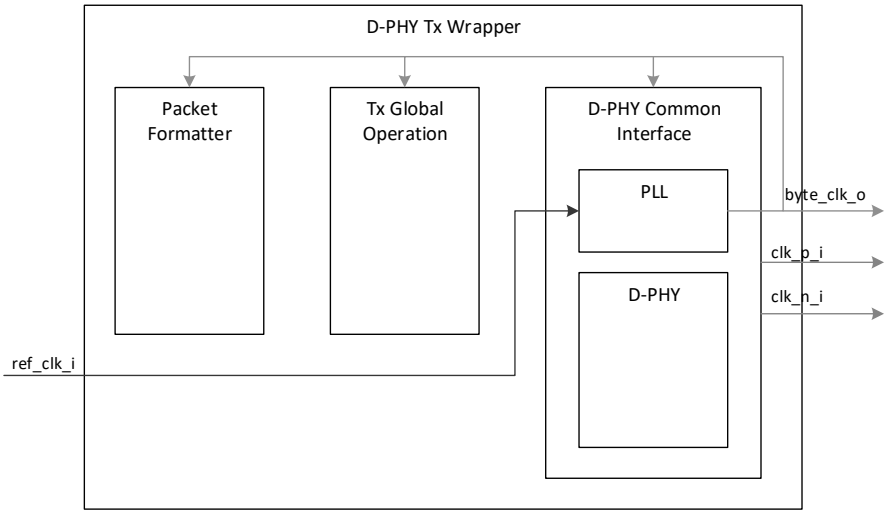

Figure 2.7. Clock Domain Crossing Block Diagram

Table 2.3. Clock Domain Crossing

Clock Domain Crossing	Handling Approach
Byte clock to DPHY bit clock	Hard D-PHY

The general formula for computing the required clocks of the system:

Tx line rate (total) $= \text{total pixels(active + blanking)} * \text{frame rate} * \text{bits per pixel}$

Tx line rate (per lane) $= \frac{\text{Tx line rate (total)}}{\text{no.of Tx lane}}$

D-PHY clock $= \frac{\text{Tx line rate (per lane)}}{2}$

Byte clock $= \frac{\text{D-PHY clock}}{\text{Tx gear}/2}$

2.2.2.1. D-PHY TX PLL

The CSI-2/DSI D-PHY Transmitter Submodule IP contains its own PLL to generate the D-PHY clock lanes and the byteclock. The block diagram of the PLL is shown in Figure 2.8.

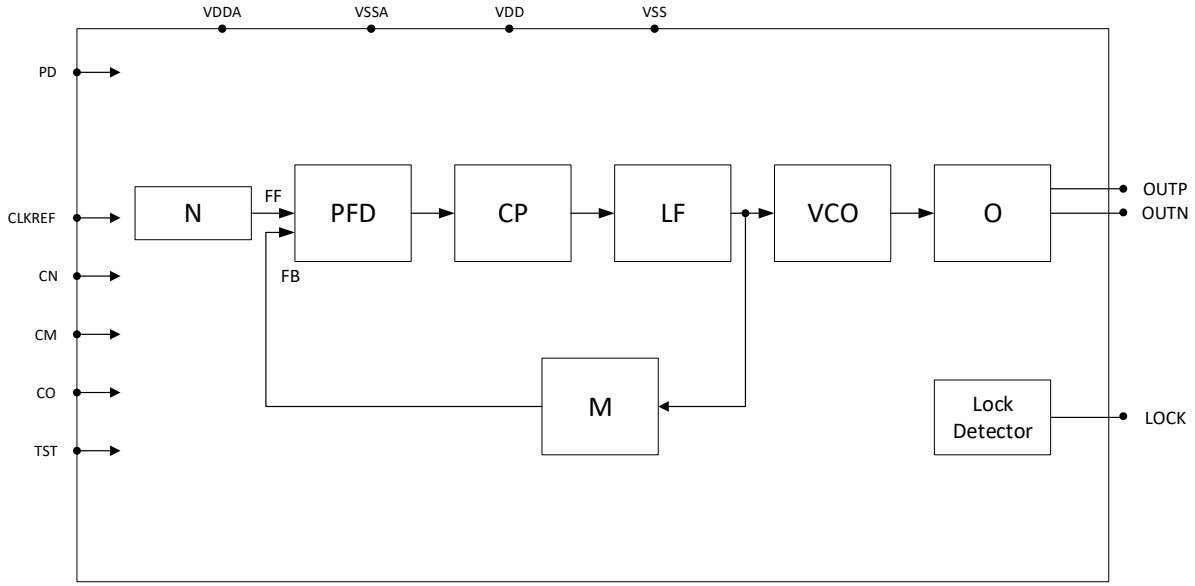

Figure 2.8. PLL Block Diagram

The output of the PLL is the D-PHY clock. It multiplies the input frequency by (M/(N*O)), where N is the input divider, M is the feedback divider, and O is the output divider.

The valid CLKREF of the D-PHY PLL, connected to the signal refclk_i, ranges from 24 MHz up to 200 MHz, but the CrossLink and CrossLinkPlus fabric supports only up to 150 MHz. The input divider, N, has to be programmed such that the frequency FF after the input divider is within 24 MHz and 30 MHz. The VCO output, which is also the input to the O divider, must be between 640 MHz and 1500 MHz.

Table 2.4. PLL Operating Frequencies

Symbol	Parameter	Min (MHz)	Max (MHz)
CLKREF	Input frequency of *refclk_i*	24	200
N	Frequency after divider N, at FF point	24	30
O	VCO output frequency, input to the O divider	640	1500

The N, M and O divider values are integer numbers and are automatically computed by the IP based on the input clock frequency and TX line rate. Due to the PLL divider requirements indicated in Table 2.4, there are certain configurations that the IP cannot support.

For the values of CN, CO, and CM, see tables in Appendix C. PLL Programming.

2.3. Design and Module Description

The top-level module instantiates the Packet Formatter, Tx Global Operation, and D-PHY Common Interface. There is an option to instantiate the internal oscillator when you do not choose an external reference clock for the D-PHY PLL.

There is also an option to disable the packet formatter module when the byte_data_i already has the header and footer.

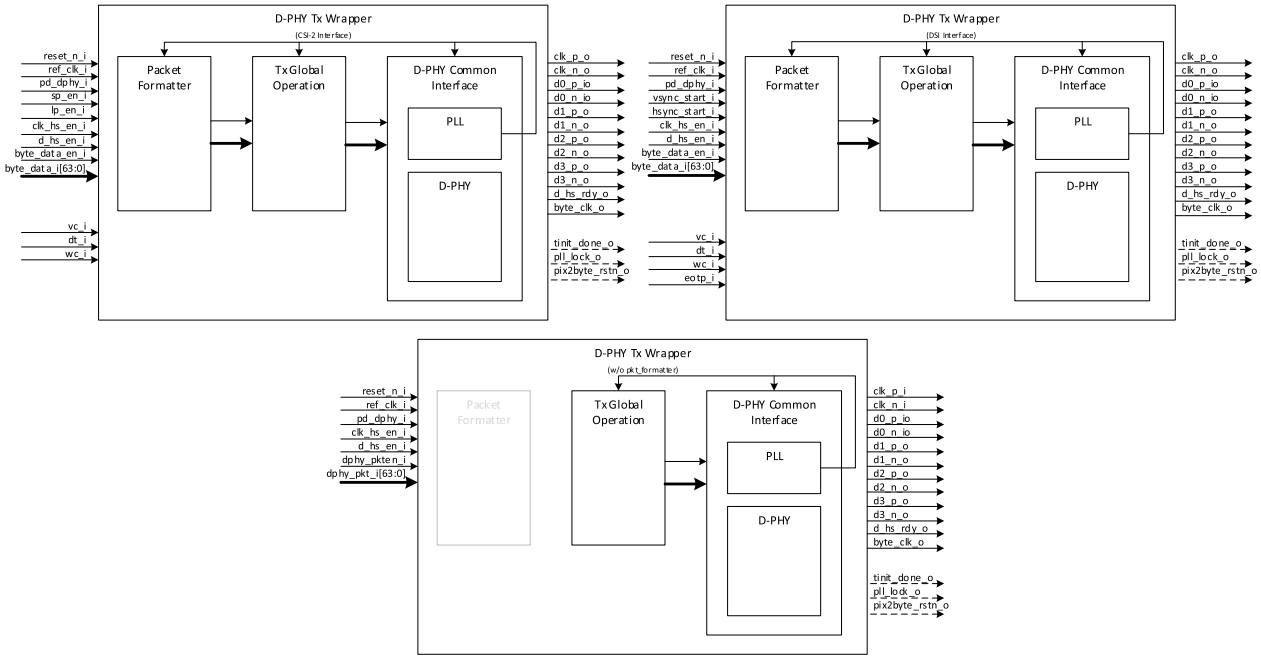

Figure 2.9. D-PHY Transmitter Submodule Block Diagram

The Packet Formatter block wraps the packet header and packet footer modules.

The Packet header module generates and appends the packet header and footer to the data payload.

The byte_data_en_i input and the byte_data_i bus are used together to identify when payload is available.

The header field and payload size is configurable by setting VC, WC, and DT parameters. End of Transmission Packet (EoTP) is supported by this module. The Packet footer module computes a CRC16 checksum based on incoming data and data enable. The data bus input is maximum 64 bits.

The Tx Global Operation block controls High Speed (HS) request path and timing using parameters. Currently Lower Power (LP) request, escape mode and turnaround path are not supported. This block follows the requirements described D-PHY Specification version 1.2 section 6 — operating modes for control and high-speed data transmission. DSI data goes into Lower-Power mode during vertical and horizontal blanking depending on the Soft IP design.

Example: HS-0/1 -> LP11(Stop) -> LP01(LP-Rqst) -> LP00(Bridge) -> HS-0/1

This module controls the timing entering HS and coming from HS entering to LP following the MIPI D-PHY Specification 1.1 Table 14. The delay parameters can be adjusted inside this module by changing its local parameter.

The DCI (D-PHY Common Interface) wrapper is used as the wrapper of MIPI D-PHY IP to make a connection between the PHY hard IP and higher protocol layers. The DCI wrapper serializes the incoming byte data and transmits it to D-PHY receiver. Based on the Tx global operation state, it determines how to enable HS or LP mode for data transfer.

3. Parameter Settings

Table 3.1 lists the parameters used to generate the D-PHY Transmitter Submodule IP. All parameters are either set automatically or input in the user interface during the D-PHY Transmitter Submodule IP generation.

Table 3.1. CSI-2/DSI D-PHY Tx Submodule IP Parameter Settings in User Interface

Parameter	Attribute	Options	Description
Number of Tx Lanes	User-Input	1, 2, 3, 4	Generate I/O up to four HS Tx data lane. 3-lane configuration is only available when the packet formatter is off.
Tx Interface	User-Input	DSI, CSI-2	Set the Tx interface.
Tx Gear	Read-Only	8, 16	Set the Tx gearing.
Target TX Line Rate	User-Input	160–1500	The target data rate for each D-PHY data lane.
Target TX Data Rate	Read-Only	<Value>	The target total data rate of the D-PHY channel. This is the target line rate multiplied by the number of data lanes.
D-PHY Clock Mode	User-Input	Continuous	In Continuous mode, the D-PHY Clock lanes are always in high speed.
		Non-continuous	In Non-continuous mode, the clock lane goes back to LP-11 state in between high speed transactions. this reduces power when the lane is idle.
Reference Clock Frequency	User-Input	24–150	Reference clock for the D-PHy PLL. Refer to the D-PHY TX PLL section for more details regarding the PLL requirements.
Generated TX Line Rate	Read-Only	<Value>	The actual line rate generated by the D-PHY PLL based on the target line rate and reference clock frequency
Generated TX Data Rate	Read-Only	<Value>	The actual line rate multiplied by the number of data lanes.
Generated D-PHY Clock Frequency	Read-Only	<Value>	The expected frequency of the D-PHY clock when in high speed mode.
Generated Byte Clock Frequency	Read-Only	<Value>	The actual line rate divided by the Tx gear.This is the clock used by the logic within the submodule IP
Deviation from Target Values	Read-Only	<Value>	Shows the deviation of the actual from the target rates.
FIFO Depth	User-Input (DSI)	Must be greater than or equal to 8	FIFO depth of the DSI packet formatter data buffer. The FIFO width is the width of the parallel data (gear*number of lanes)
FIFO Type	User-Input (DSI)	EBR, LUT	FIFO implementation of the DSI packet formatter data buffer.
tINIT_SLAVE Value	User-Input	<Value>	Specify delay time for Tx D-PHY tINIT requirement. Must satisfy Tx D-PHY tINIT minimum requirement of 100 µs and is clocked by byte clock.
Disable tINIT counter	User-Input	Enable Disable	Enable or disable tINIT counter for resource utilization.
Disable Packet formatter	User-Input	Enable Disable	Enable or disable packet formatter module.
Enable Frame number increment	User-Input	Enable Disable	Enable or disable frame number increment in packet formatter for CSI-2 interface.
Enable Line number increment	User-Input	Enable Disable	Enable or disable line number increment in packet formatter for CSI-2 interface.

Parameter	Attribute	Options	Description
TLPX Clock Prepare Clock HS-zero Clock Pre Clock Post Clock Trail Clock Exit Data Prepare Data HS-Zero Data Trail Data Exit	User-Input	<Value>	D-PHY Global Operation Timing parameters. Refer to section 6.9 of MIPI D-PHY Specification v1.1.

4. IP Generation and Evaluation

This section provides information on how to generate the Lattice D-PHY Transmitter Submodule IP code using the Lattice Diamond Clarity Designer and how to run simulation, synthesis, and hardware evaluation.

4.1. Licensing the IP

The D-PHY Transmitter IP is available free of charge, but an IP-specific license is required to enable full, unrestricted use of the D-PHY Transmitter IP in a complete, top level design.

Request your license by going to the link http://www.latticesemi.com/en/Support/Licensing and request the free Lattice Diamond license. In this form, select the desired CrossLink/CrossLinkPlus IP for your design.

You may download and generate the D-PHY Transmitter IP and fully evaluate the IP through functional simulation and implementation (synthesis, map, place and route) without an IP license. The D-PHY Transmitter IP also supports Lattice IP hardware evaluation capability, see the Hardware Evaluation section for further details.

HOWEVER, THE IP LICENSE IS REQUIRED TO ENABLE TIMING SIMULATION, TO OPEN THE DESIGN IN DIAMOND EPIC TOOL, OR TO GENERATE BITSTREAMS THAT DO NOT INCLUDE THE HARDWARE EVALUATION TIMEOUT LIMITATION.

4.2. Getting Started

The D-PHY Transmitter IP is available for download from the Lattice IP Server using the Clarity Designer tool. The IP files are automatically installed using ispUPDATE technology in any customer-specified directory. After the IP is installed, the IP is available in the Clarity Designer user interface as shown in Figure 4.1.

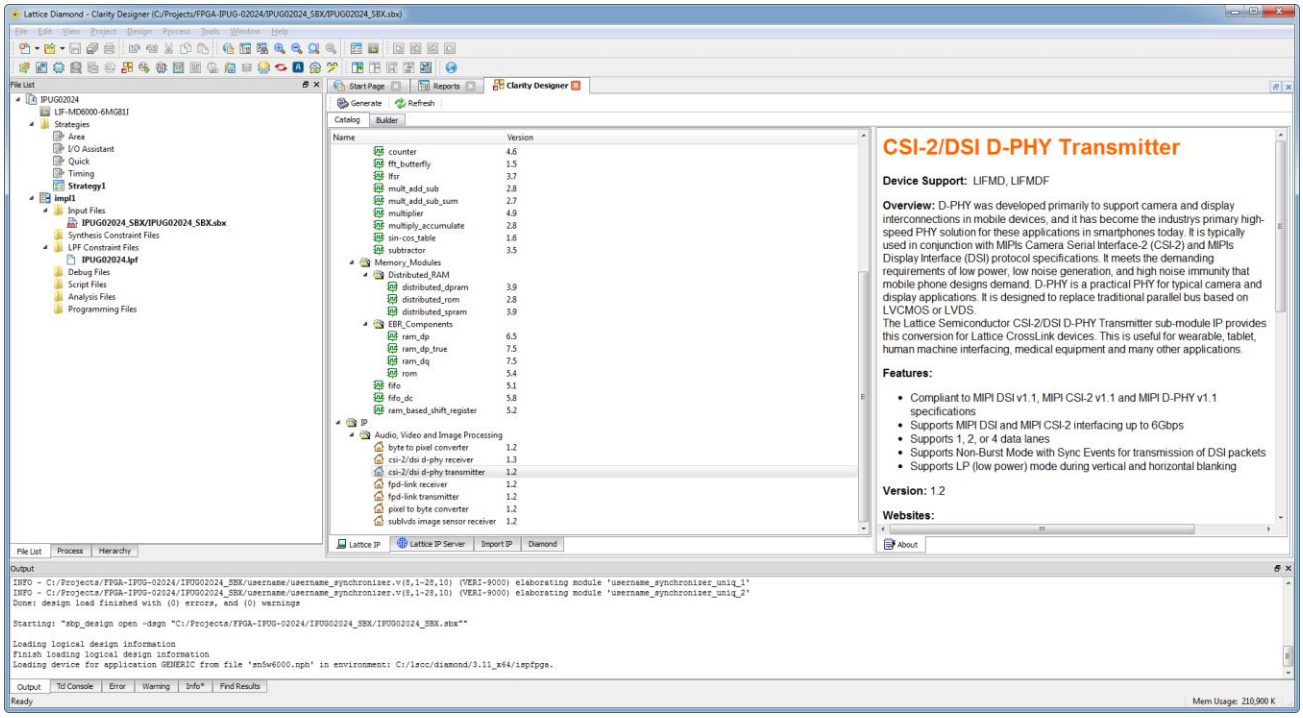

Figure 4.1. Clarity Designer Window

4.3. Generating IP in Clarity Designer

The Clarity Designer tool is used to customize modules and IPs and place them into the device architecture. Besides configuration and generation of modules and IPs, Clarity Designer can also create a top module template in which all generated modules and IPs are instantiated.

The procedure for generating D-PHY Transmitter IP in Clarity Designer is described below.

Clarity Designer can be started from the Diamond design environment.

To start Clarity Designer:

1. Create a new Diamond project for CrossLink or CrossLinkPlus family devices.
2. From the Diamond main window, choose **Tools > Clarity Designer**, or click  in Diamond toolbox. The **Clarity Designer** project dialog box is displayed.
3. Select and/or fill out the following items as shown in Figure 4.2.
   - **Create new Clarity design** – Select this to create a new Clarity Design project directory in which the D-PHY Transmitter IP is generated.
   - **Design Location** – Clarity Design project directory path.
   - **Design Name** – Clarity Design project name.
   - **HDL Output** – Hardware Description Language Output Format (Verilog HDL).

   The Clarity Designer project dialog box also allows you to open an existing Clarity Designer project by selecting the following:
   - **Open Clarity design** – Open an existing Clarity Design project.
   - **Design File** – Name of existing Clarity Design project file with .sbx extension.
4. Click the **Create** button. A new Clarity Designer project is created.

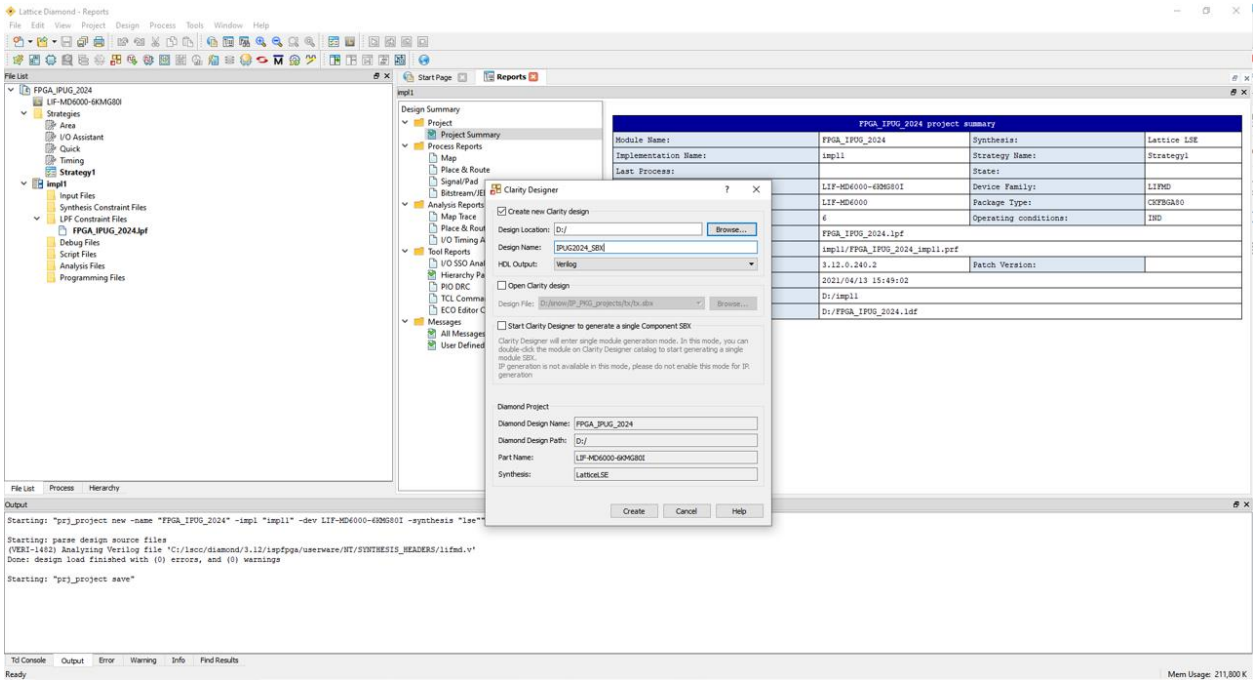

Figure 4.2. Starting Clarity Designer from Diamond Design Environment

To configure D-PHY Transmitter IP in Clarity Designer:

1. Double-click **csi-2/dsi d-phy transmitter** in the IP list of the System Catalog view. The **csi-2/dsi d-phy transmitter** dialog box is displayed as shown in Figure 4.3.

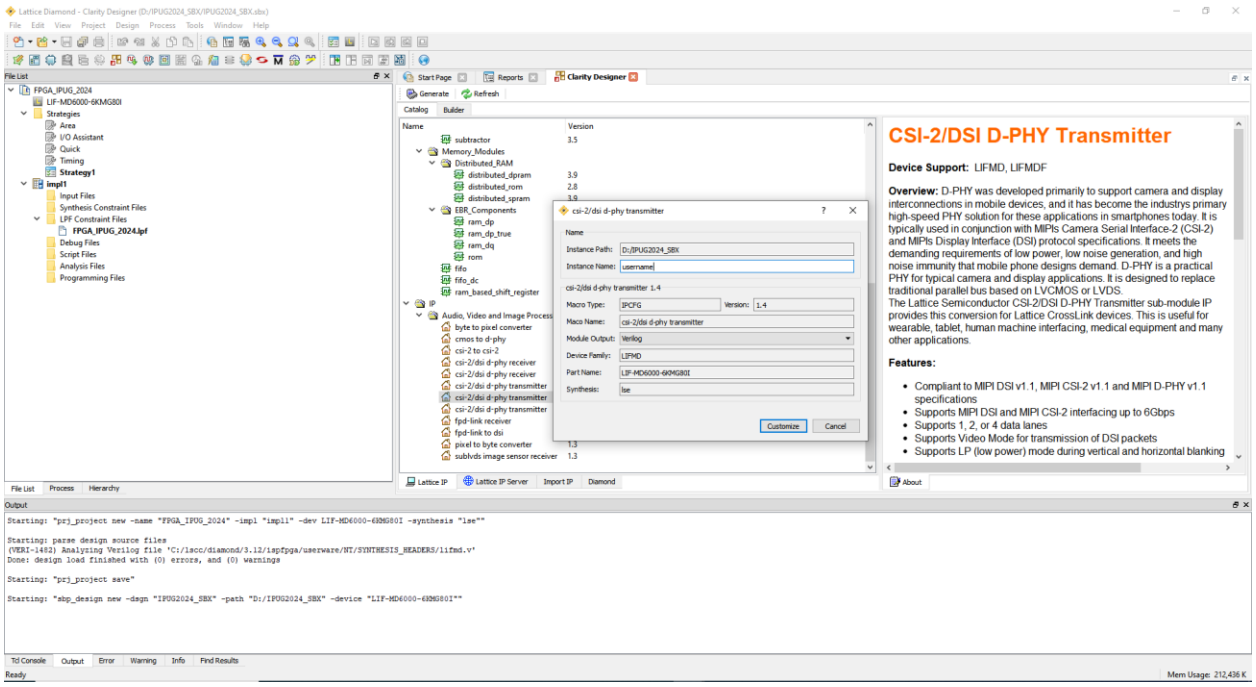

Figure 4.3. Configuring D-PHY Transmitter IP in Clarity Designer

2. Enter the **Instance Name**.
3. Click the **Customize** button. An IP configuration interface is displayed as shown in Figure 4.4. From this dialog box, you can select the IP configuration specific to your application.
4. Input valid values in the required fields in the **Configuration** tab.

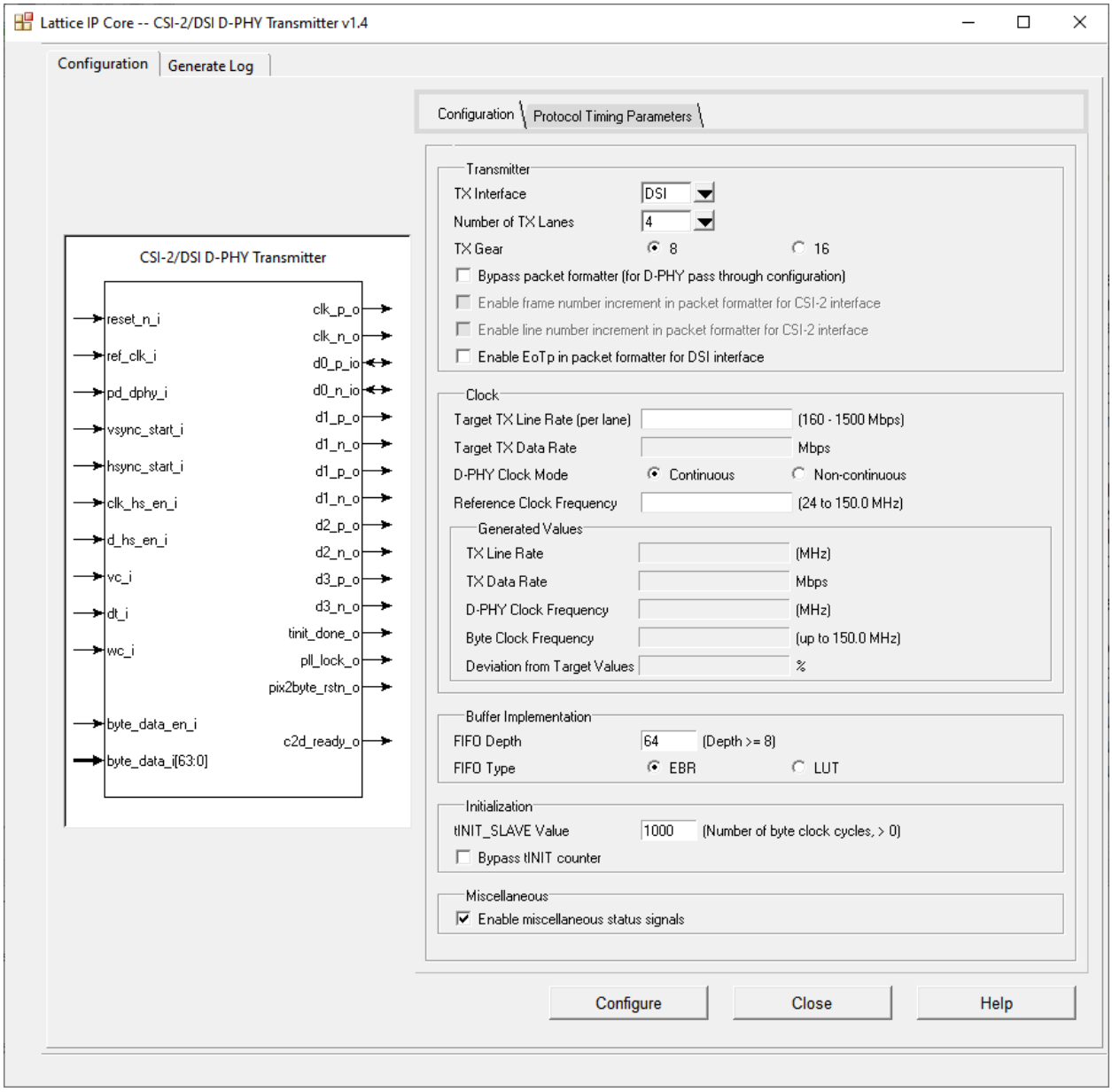

Figure 4.4. Configuration Tab in IP User Interface

5. Go to **Protocol Timing Parameters** tab. The values shown are automatically computed. You can enter other valid values if customization is desired.
6. After selecting the required parameters, click the **Configure** button.
7. Click **Close**.
8. Click Generate in the toolbox. Clarity Designer generates all the IPs and modules, and creates a top module to wrap them.

For detailed instructions on how to use the Clarity Designer, refer to the Lattice Diamond software user guide.

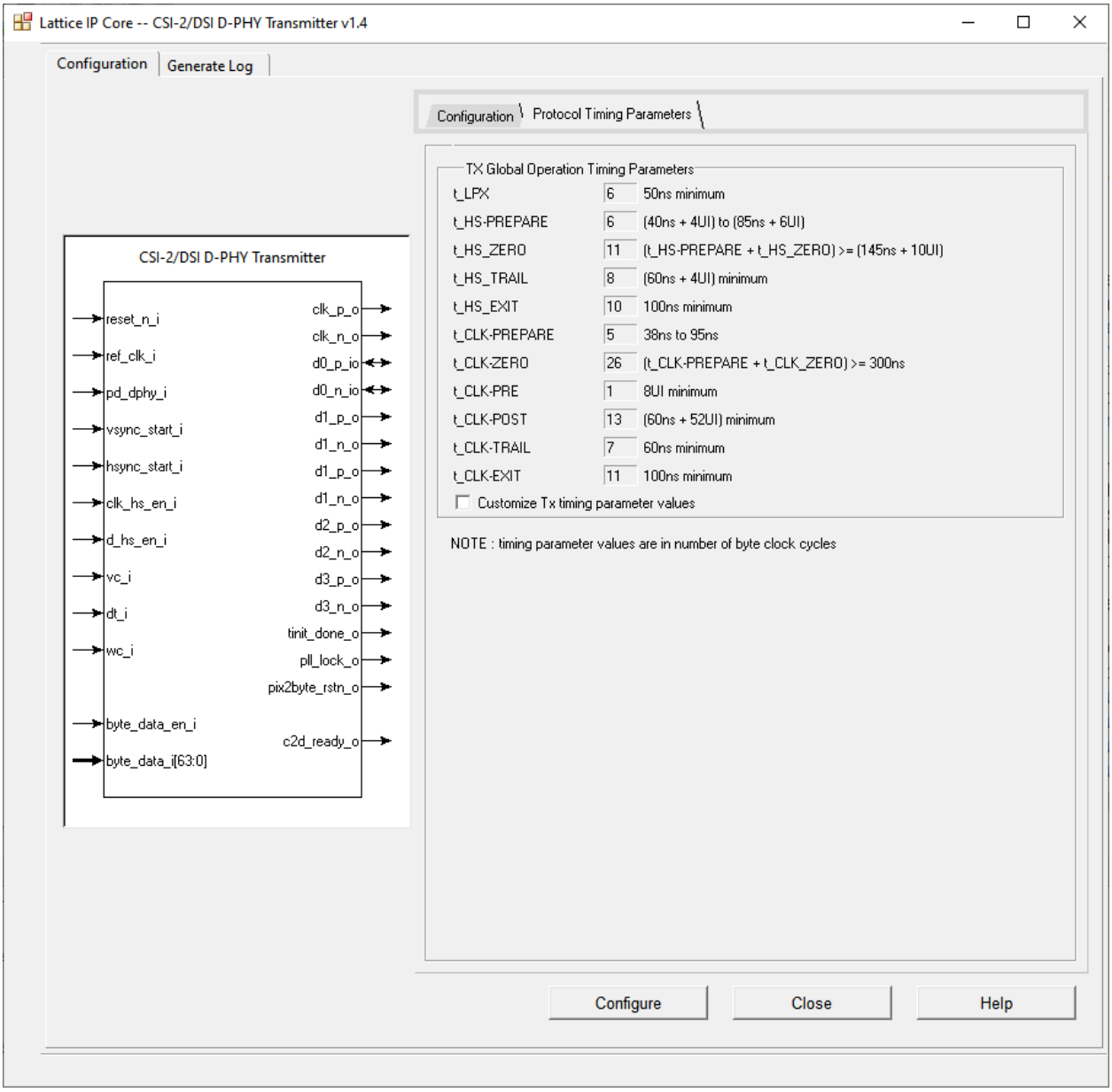

Figure 4.5. Protocol Timing Parameter Tab in IP User Interface

4.4. Generated IP Directory Structure and Files

Figure 4.6 shows the directory structure of generated IP and supporting files.

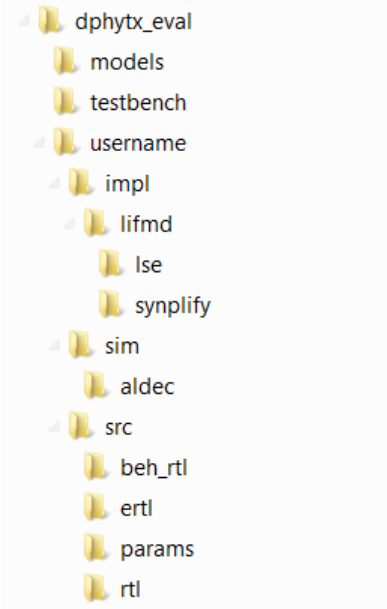

Figure 4.6. D-PHY Transmitter IP Directory Structure

The design flow for the IP created with Clarity Designer uses a post-synthesized module (NGO) for synthesis and a protected model for simulation. The post-synthesized module and protected model are customized when you configure the IP and created automatically when the IP is generated.

Table 4.1 provides a list of key files and directories created by Clarity Designer and how they are used. The post-synthesized module (NGO), the protected simulation model, and all other files are also generated based on your configuration and are provided as examples to use or evaluate the IP.

Table 4.1. Files Generated in Clarity Designer

File	Description
<instance_name>.v	Verilog top-level module of D-PHY Transmitter IP used for both synthesis and simulation.
*<instance_name>*_*.v	Verilog submodules for simulation. Files that do not have equivalent black box modules are also used for synthesis.
*<instance_name>*_*_beh.v	Protected Verilog models for simulation.
*<instance_name>*_*_bb.v	Verilog black box modules for synthesis.
*<instance_name>*_*.ngo	User interface configured and synthesized modules for synthesis.
*<instance_name>*_params.*v*	Verilog parameters file that contains required compiler directives to successfully configure IP during synthesis and simulation.
<instance_name>.lpc	Lattice Parameters Configuration file. This file records all the IP configuration options set through Clarity Designer. It is used by IP generation script to generate configuration-specific IP. It is also used to reload parameter settings in the IP user interface in Clarity Designer when it is being reconfigured.
*<instance_name>*_wrap.v	Evaluation top-level module for reduced number of pinlist. Use this as the top-level module for Map and PAR evaluation purposes only.
*<instance_name>*_inst.v/vhd	Template for instantiating the generated soft IP top-level in another user-created top module.

Aside from the files listed in the tables, most of the files required to evaluate the D-PHY Transmitter IP are available under the directory *\<dphytx_eval>*. This includes the simulation model. Lattice Diamond project files are also included under the folder at *\<dphytx_eval>\<instance_name>\impl\<device_family>\<synthesis_tool>*, where *<device_family>* can either be **lifmd** for CrossLink or **lifmdf** for CrossLinkPlus devices.

The \<*instance_name*> folder contains files/folders with content specific to the <*instance_name*> configuration. This directory is created by Clarity Designer each time the IP is generated and regenerated with the same file name. A separate \<*instance_name*> directory is generated for IPs with different names, such as \<*my_IP_0*>,\<*my_IP_1*>, and others.

The folder\<*instance_name*>, the *dphytx_eval* and sub directories provide files supporting D-PHY Transmitter IP evaluation that includes files/folders with content that is constant for all configurations of the D-PHY Transmitter IP. The *dphytx_eval* directory is created by Clarity Designer the first time the IP is generated, when multiple D-PHY Transmitter IPs are generated in the same root directory and updated each time the IP is regenerated.

You can use the prebuilt Diamond projects provided at \<*project_root*>*dphytx_eval*\<*instance_name*>*impl*\<*device_family*>\<*synthesis_tool*>\ to evaluate the implementation (synthesis, map, place and route) of the IP in Lattice Diamond tool. This replaces the <instance_name>.v with the dummy evaluation module <instance_name>_wrap.v to reduce the pin list so that Map and PAR pass. The *src* directory contains the behavioral models of the black-boxed modules and the *models* directory provides library elements.

4.5. Running Functional Simulation

To run simulations using Active-HDL:

1. Under the **Tools** menu in Diamond, select **Active-HDL**.
2. In **Active-HDL** window, under the **Tools** tab, select **Execute Macro**.
3. Select the **.do** file **\<project_dir>\dphytx_eval\<instance_name>\sim\aldec*_run.do**.
4. Click **OK**.
5. Wait for simulation to finish.
6. To override default TB parameters, modify the **\<project_dir>\dphytx_eval\testbench\tb_setup_params.v** file.

Table 4.2. Testbench Compiler Directives

Compiler Directive	Description
NUM_BYTES	Number of bytes per line.
NUM_LINES	Number of lines per frame.
NUM_FRAMES	Number of frames to be transmitted.
DATA_TYPE	6-bits value of data type to be transmitted.
DEBUG_ON	Enable debug messages.
EOTP_ENABLE	Enable EoTP in DSI mode.
LS_LE_EN	Enable model transmission of LS and LE short packet in CSI-2 mode.
VIRTUAL_CHANNEL	Set virtual channel.
HSYNC_PULSE_FRONT	Number of HSYNC pulses before data transmission.
HSYNC_PULSE_BACK	Number of HSYNC pulses after data transmission.
HS_RDY_NEG_TO_HS_CLK_EN_DLY	Delay from d_hs_en negation to next d_hs_rdy_o assertion.
HS_RDY_TO_BYTE_DATA_EN_DLY	Delay from d_hs_en assertion to byte_data_en assertion.
HS_RDY_TO_DPHY_PKTEN_DLY	Delay from d_hs_en assertion to dphy_pkten assertion.
HS_RDY_TO_HSYNC_START_DLY	Delay from d_hs_en assertion to hsync_start assertion.
HS_RDY_TO_VSYNC_START_DLY	Delay from d_hs_en assertion to vsync_start assertion.
HS_RDY_TO_LP_EN_DLY	Delay from d_hs_en assertion to lp_en assertion.
HS_RDY_TO_SP_EN_DLY	Delay from d_hs_en assertion to sp_en assertion.
LP_EN_TO_BYTE_DATA_EN_DLY	Delay from lp_en pulse to byte_data_en assertion.
HSYNC_TO_HSYNC_DLY	Delay from hsync to next hsync packet transmission.
VSYNC_TO_HSYNC_DLY	Delay from vsync to next hsync packet transmission.
TINIT_DURATION	Used when MISC_ON is not defined. This is for setting the wait time duration (in ps) of tINIT ROM done.

4.6. Simulation Strategies

This section describes the simulation environment, which demonstrates basic D-PHY Transmitter functionality. Figure 4.7 shows the block diagram of simulation environment.

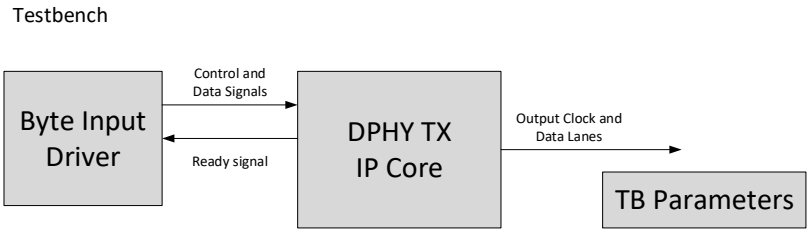

Figure 4.7. Simulation Environment Block Diagram

4.7. Simulation Environment

The simulation environment is made up of an input driver instance connected to the input of CSI-2/DSI D-PHY Transmitter IP Core instance in the testbench. The input driver is configured based on CSI-2/DSI D-PHY Transmitter IP Core configurations and testbench parameters. It can be configured to drive data up to 8 bytes per clock cycle depending on the number of D-PHY Tx lanes and Tx gearing.

The input driver can also be configured to drive control signals specific to DSI and CSI-2. If miscellaneous signals are enabled, the testbench waits for tinit_done assertion before transmitting byte data. Otherwise, a user-configurable delay (tinit_duration) can be used to ensure that initialization is done before testbench transmits byte data. Refer to the *tb_setup_params.v* file for different testbench parameters that can be used to control the simulation.

Figure 4.8 shows the CSI-2/DSI D-PHY Transmitter IP core configured as CSI-2 transmitter. Packet transmissions are controlled by driving sp_en for short packets such as frame start, frame end, line start and line end packets. The lp_en signal is used to drive data packets together with byte_data_en, byte_en and other control signals.

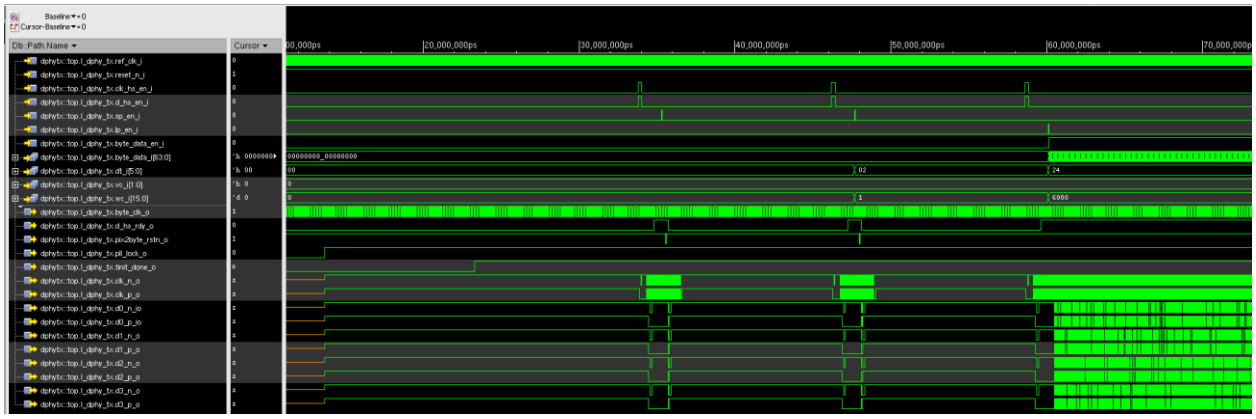

Figure 4.8 D-PHY Tx configured as CSI-2 Transmitter

Figure 4.9 shows the D-PHY Transmitter IP core configured as DSI transmitter. Sync packets are controlled by driving vsync_start, hsync_start signals. Data transmissions are controlled by driving byte_data_en, byte_data input bus with other control signals.

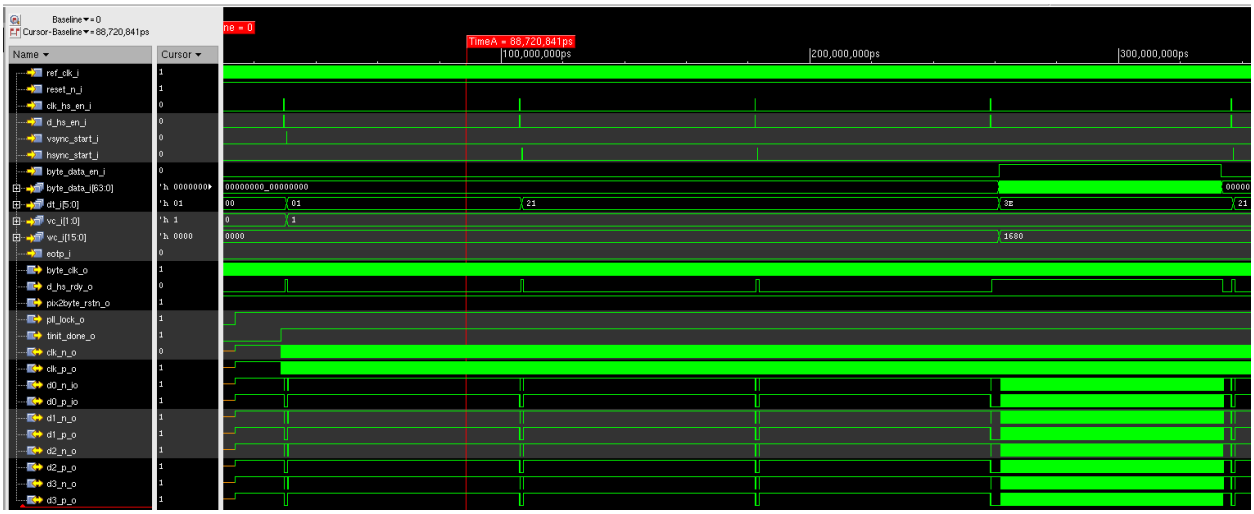

Figure 4.9. D-PHY Tx configured as DSI Transmitter

4.8. Instantiating the IP

The core modules of the CSI-2/DSI D-PHY Transmitter IP are synthesized and provided in NGO format with black box Verilog source files for synthesis. A Verilog source file named *<instance_name>_dphy_tx.v* instantiates the black box of core modules. The top-level file *<instance_name>.v* instantiates *<instance_name>_dphy_tx.v*.

A Verilog instance template *<instance_name>_inst.v* or VHDL instance template *<instance_name>_inst.vhd* is also provided as a guide if the design is to be included in another top level module.

You do not need to instantiate the IP instances one by one manually. The top-level file and other Verilog source files are provided in *\<project_dir>*. These files are refreshed each time the IP is regenerated.

4.9. Synthesizing and Implementing the IP

In Clarity Designer, the Clarity Designer project file (.sbx) is added to Lattice Diamond as a source file after all IPs are generated. Note that default Diamond strategy (.sty) and default Diamond preference file (.lpf) are used. When using the .sbx approach, import the recommended strategy and preferences from *\<project_dir>\dphytx_eval\<instance_name>\impl\<device_family>\lse* or *\<project_dir>\dphytx_eval\<instance_name>\impl\<device_family>\synplify* directories. All required files are invoked automatically. You can directly synthesize, map and place/par the design in the Diamond design environment after the cores are generated.

Push-button implementation of this top-level design with either Synplify or Lattice Synthesis Engine is supported via the Diamond project files *<instance_name>_top.ldf*, which is located in the *\<project_dir>\dphytx_eval\<instance_name>\impl\<device_family>\<synthesis_tool>* directory.

To use the pre-built Diamond project files:

1. Choose **File > Open > Project**.
2. In the **Open Project** dialog box, browse to **\<project_dir>\dphytx_eval\<instance_name>\impl\<device_family>\<synthesis_tool>**.
3. Select and open **<instance_name>_top.ldf**. At this point, all of the files needed to support top-level synthesis and implementation are imported to the project.
4. Select the **Process** tab in the left-hand user interface window.
5. Implement the complete design via the standard Diamond user interface flow.

4.10. Hardware Evaluation

The CSI-2/DSI D-PHY Transmitter IP supports Lattice IP hardware evaluation capability. You can create versions of the IP that operate in hardware for a limited period of time without requiring the request of an IP license. It may also be used to evaluate the IP in hardware in user-defined designs.

4.10.1. Enabling Hardware Evaluation in Diamond

Choose **Project** > **Active Strategy** > **Translate Design Settings**. The hardware evaluation capability may be enabled or disabled in the **Strategy** dialog box. It is enabled by default.

4.11. Updating/Regenerating the IP

The Clarity Designer interface allows you to update the local IPs from the Lattice IP server. The updated IP can be used to regenerate the IP in the design. To change the parameters of the IP used in the design, the IP must also be regenerated.

4.11.1. Regenerating an IP in Clarity Designer

To regenerate IP in Clarity Designer:

1. In the **Builder** tab, right-click the IP instance to be regenerated and select **Config** from the menu as shown in Figure 4.10.

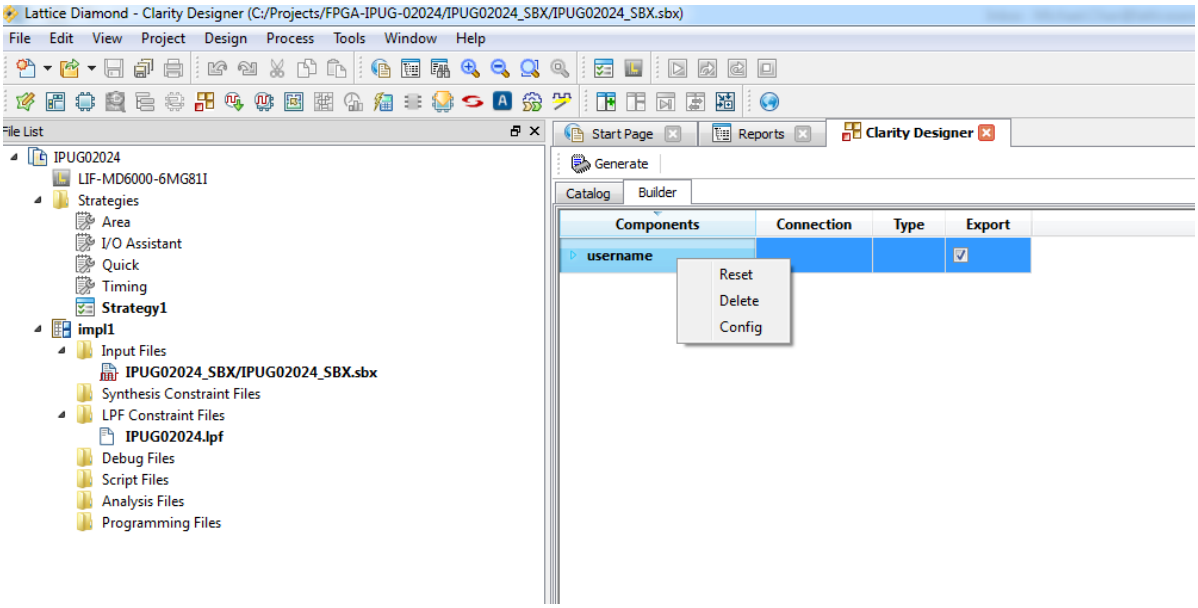

Figure 4.10. IP Regeneration in Clarity Designer

2. The IP Configuration user interface is displayed. Change the parameters as required and click the **Configure** button.
3. Click Generate in the toolbox. Clarity Designer regenerates all the instances which are reconfigured.

References

For more information about CrossLink and CrossLinkPlus devices, refer to CrossLink Family Data Sheet (FPGA-DS-02007) and CrossLinkPlus Family Data Sheet (FPGA-DS-02054).

Software documentation:

- Clarity Designer User Manual
- Lattice Diamond User Guide

For further information on interface standards, refer to:

- MIPI Alliance Specification for D-PHY, version 1.1, November 7, 2011, www.mipi.org
- MIPI Alliance Specification for Display Serial Interface, version 1.1, November 22, 2011, www.mipi.org
- MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2), version 1.1, July 18, 2012, www.mipi.org

Technical Support Assistance

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

Appendix A. Resource Utilization

Table A.1 lists resource utilization information for Lattice CrossLink FPGA using the CSI-2/DSI D-PHY Transmitter IP.

Clarity Designer is the Lattice IP configuration utility, and is included as a standard feature of the Diamond tool. For details about the usage of Clarity Designer, refer to the Clarity Designer and Diamond help system.

For more information on the Diamond design tools, visit the Lattice web site at www.latticesemi.com/Products/DesignSoftwareAndIP.

Table A.1. Resource Utilization[1, 3]

IP User-Configurable Parameters	Registers	Slices	LUTs	sysMEM EBRs	MIPI D-PHY	Target f_{MAX} (MHz)[2]	Actual f_{MAX} (MHz)[2]
CSI-2 Gear 16							
4-lane Non-continuous clock	517	987	1496	0	1	93.75	114.71
2-lane Non-continuous clock	364	605	936	0	1	93.75	103.91
1-lane Non-continuous clock	309	484	758	0	1	93.75	133.35
4-lane Continuous clock	509	937	1433	0	1	93.75	122.91
2-lane Continuous clock	356	561	873	0	1	93.75	108.86
1-lane Continuous clock	301	443	695	0	1	93.75	129.87
CSI-2 Gear 8							
4-lane Non-continuous clock	519	648	989	0	1	150	152.55
2-lane Non-continuous clock	336	452	699	0	1	150	154.27
1-lane Non-continuous clock	276	385	587	0	1	150	149.99
4-lane Continuous clock	511	620	947	0	1	150	152.93
2-lane Continuous clock	328	424	657	0	1	150	153.61
1-lane Continuous clock	268	357	545	0	1	150	150.83
DSI Gear 16							
4-lane Non-continuous clock	656	2901	5548	0	1	93.75	109
2-lane Non-continuous clock	438	1617	2960	0	1	93.75	132.2
1-lane Non-continuous clock	334	937	1683	0	1	93.75	117.84
4-lane Continuous clock	648	2856	5485	0	1	93.75	109.23
2-lane Continuous clock	430	1573	2897	0	1	93.75	135.46
1-lane Continuous clock	326	896	1620	0	1	93.75	128.09

IP User-Configurable Parameters	Registers	Slices	LUTs	sysMEM EBRs	MIPI D-PHY	Target f_{MAX} (MHz)[2]	Actual f_{MAX} (MHz)[2]
DSI Gear 8							
4-lane Non-continuous clock	659	2676	5125	0	1	112.5	118
2-lane Non-continuous clock	447	1546	2836	0	1	150	157.11
1-lane Non-continuous clock	331	834	1513	0	1	93.75	120.12
4-lane Continuous clock	651	2653	5083	0	1	93.75	103.85
2-lane Continuous clock	439	1518	2794	0	1	150	150.92
1-lane Continuous clock	323	806	1471	0	1	112.5	128.32

Notes:

1. Performance and utilization data target an LIF-MD6000-6MG81I device using Lattice Diamond 3.9 and Lattice Synthesis Engine software. Performance may vary when using a different software version or targeting a different device density or speed grade within the CrossLink family. This does not show all possible configurations of the D-PHY Transmitter IP.
2. The f_{MAX} values are based on byte clock and may vary depending on the complete top level-design.
3. 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. What is Not Supported

The IP does not support configuration through registers.

Appendix C. PLL Programming

Table C.1. 5-Bit Input Divider

N	CN[4:0]
1	11111
2	00000
3	10000
4	11000
5	11100
6	01110
7	00111
8	10011
9	01001
10	00100
11	00010
12	10001
13	01000
14	10100
15	01010
16	10101
17	11010
18	11101
19	11110
20	01111
21	10111
22	11011
23	01101
24	10110
25	01011
26	00101
27	10010
28	11001
29	01100
30	00110
31	00011
32	00001

Table C.2. 8-Bit Feedback Divider

DVR	CM[7:0]	DVR	CM[7:0]	DVR	CM[7:0]	DVR	CM[7:0]	DVR	CM[7:0]
16	111X0000	64	10000000	112	10110000	160	00100000	208	01010000
17	111X0001	65	10000001	113	10110001	161	00100001	209	01010001
18	111X0010	66	10000010	114	10110010	162	00100010	210	01010010
19	111X0011	67	10000011	115	10110011	163	00100011	211	01010011
20	111X0100	68	10000100	116	10110100	164	00100100	212	01010100
21	111X0101	69	10000101	117	10110101	165	00100101	213	01010101
22	111X0110	70	10000110	118	10110110	166	00100110	214	01010110
23	111X0111	71	10000111	119	10110111	167	00100110	215	01010111
24	111X1000	72	10001000	120	10111000	168	00101000	216	01011000
25	111X1001	73	10001001	121	10111001	169	00101001	217	01011001
26	111X1010	74	10001010	122	10111010	170	00101010	218	01011010
27	111X1011	75	10001011	123	10111011	171	00101011	219	01011011
28	111X1100	76	10001100	124	10111100	172	00101100	220	01011100
29	111X1101	77	10001101	125	10111101	173	00101101	221	01011101
30	111X1110	78	10001110	126	10111110	174	00101110	222	01011110
31	111X1111	79	10001111	127	10111111	175	00101111	223	01011111
32	11000000	80	10010000	128	00000000	176	00110000	224	01100000
33	11000001	81	10010001	129	00000001	177	00110001	225	01100001
34	11000010	82	10010010	130	00000010	178	00110010	226	01100010
35	11000011	83	10010011	131	00000011	179	00110011	227	01100011
36	11000100	84	10010100	132	00000100	180	00110100	228	01100100
37	11000101	85	10010101	133	00000101	181	00110101	229	01100101
38	11000110	86	10010110	134	00000110	182	00110110	230	01100110
39	11000111	87	10010111	135	00000111	183	00110111	231	01100111
40	11001000	88	10011000	136	00001000	184	00111000	232	01101000
41	11001001	89	10011001	137	00001001	185	00111001	233	01101001
42	11001011	90	10011010	138	00001010	186	00111010	234	01101010
43	11001011	91	10011011	139	00001011	187	00111011	235	01101011
44	11001100	92	10011100	140	00001100	188	00111100	236	01101100
45	11001101	93	10011101	141	00001101	189	00111101	237	01101101
46	11001110	94	10011110	142	00001110	190	00111110	238	01101110
47	11001111	95	10011111	143	00001111	191	00111111	239	01101111
48	11010000	96	10100000	144	00010000	192	01000000	240	01110000
49	11010001	97	10100001	145	00010001	193	01000001	241	01110001
50	11010010	98	10100010	146	00010010	194	01000010	242	01110010
51	11010011	99	10100011	147	00010011	195	01000011	243	01110011
52	11010100	100	10100100	148	00010100	196	01000100	244	01110100
53	11010101	101	10100101	149	00010101	197	01000101	245	01110101
54	11010110	102	10100110	150	00010110	198	01000110	246	01110110
55	11010111	103	10100111	151	00010111	199	01000111	247	01110111
56	11011000	104	10101000	152	00011000	200	01001000	248	01111000
57	11011001	105	10101001	153	00011001	201	01001001	249	01111001
58	11011010	106	10101010	154	00011010	202	01001010	250	01111010
59	11011011	107	10101011	155	00011011	203	01001011	251	01111011
60	11011100	108	10101100	156	00011100	204	01001100	252	01111100
61	11011101	109	10101101	157	00011101	205	01001101	253	01111101
62	11011110	110	10101110	158	00011110	206	01001110	254	01111110

DVR	CM[7:0]	DVR	CM[7:0]	DVR	CM[7:0]	DVR	CM[7:0]	DVR	CM[7:0]
63	11011111	111	10101111	159	00011111	207	01001111	255	01111111

Table C.3. 2-Bit Output Divider

O	CO1	CO0
1	0	0
2	0	1
4	1	0
8	1	1

Revision History

Revision 1.6, IP Version 1.4, April 2021

Section	Change Summary
Acronyms in This Document	Added this section.
Introduction	Updated IP design version to 1.4. Updated burst mode, non-burst mode feature.
Functional Description	• Updated steps in the Input Interface section. • Added Packet Transmission when Packet Formatter is Disabled heading and moved contents under this section. • Added reference to Table 2.2. Byte Data Bus Assignments. • Updated statement in the Reset and Initialization section as follows: *When set in DSI, until the DCS commands are sent to the display, valid data from the Application Processor may be lost*. • Corrected *LS mode* to *LP mode* in the Design and Module Description section.
Parameter Settings	Updated Table 3.1. CSI-2/DSI D-PHY Tx Submodule IP Parameter Settings in User Interface.
IP Generation and Evaluation	Updated Figure 4.2 to Figure 4.5.

Revision 1.5, IP Version 1.3, April 2021

Section	Change Summary
Functional Description	Added reference to Appendix C. PLL Programming tables in D-PHY TX PLL section.
Appendix C. PLL Programming	Added this section.

Revision 1.4, IP Version 1.3, July 2020

Section	Change Summary

Introduction	• Updated Table 1.1. CSI-2/DSI D-PHY Transmitter Submodule IP Quick Facts.
	  • Updated LUTs and Registers values.
	  • Updated note.

Functional Description	Corrected text alignment in Figure 2.9. D-PHY Transmitter Submodule Block Diagram.
Parameter Settings	Added DPHY Clock Mode parameter in Table 3.1. CSI-2/DSI D-PHY Tx Submodule IP Parameter Settings in User Interface.
Appendix A. Resource Utilization	Updated Table A.1. Resource Utilization1.

Revision 1.3, IP Version 1.3, March 2020

Section	Change Summary

Introduction	• Changed IP design version from 1.x to 1.3.
	• Updated Table 1.1. CSI-2/DSI D-PHY Transmitter Submodule IP Quick Facts.
	  • Updated LUTs and Registers values.
	  • Updated note.
	• Changed feature to *Supports 1, 2, 3, or 4 MIPI D-PHY data lanes*.
	• Update number of Tx lanes options in Table 1.2. CSI-2/DSI D-PHY Transmitter Submodule IP Features Summary.

Functional Description	• Added frame_max_i in Figure 2.1. CSI-2/DSI D-PHY Transmitter Submodule Top-level Block Diagram.
	• Updated Table 2.1. D-PHY Transmitter Submodule IP Pin Function Description .
	• Updated the Short Packet Transmission and the Long Packet Transmission procedures.
	• Updated the following figures:
	  • Figure 2.2. D-PHY Tx Input Bus for Short Packet Transmission
	  • Figure 2.3. D-PHY Tx Input Bus for Long Packet Transmission in CSI-2 Interface
	  • Figure 2.4. D-PHY Tx Input Bus for Long Packet Transmission in DSI Interface
	  • Figure 2.5. D-PHY Tx Input Bus for Long Packet Transmission in DSI/CSI-2 Interface without pkt_formatter
	• Updated the following sections:
	  • Output Interface
	  • Clock Domains and Clock Domain Crossing
	  • Reset and Initialization
	• Updated Table 2.4. PLL Operating Frequencies.
	• Added D-PHY TX PLL section.
	• Removed Compiler Directives section.

Parameter Settings	• Updated section header. • Updated Table 3.1. CSI-2/DSI D-PHY Tx Submodule IP Parameter Settings in User Interface.

IP Generation and Evaluation	Updated the following figures:
	  • Figure 4.4. Configuration Tab in IP User Interface
	  • Figure 4.5. Protocol Timing Parameter Tab in IP User Interface
	• Added <instance_name>_wrap.v in Table 4.1. Files Generated in Clarity Designer.
	• Added context information.

Appendix A. Resource Utilization	Updated Table A.1. Resource Utilization.
—	Minor editorial changes

Revision 1.2, IP Version 1.3, October 2019

Section	Change Summary
Disclaimer	Newly added section.
All	Added CrossLinkPlus device support. Minor adjustments in style and formatting.
References	Updated.

Revision 1.1, IP Version 1.0, April 2019

Section	Change Summary
Introduction	Specified that this user guide can be used for IP design versions 1.x.
IP Generation and Evaluation	In Licensing the IP, modified the instructions for requesting free license.
Revision History	Updated revision history table to new template.
All	Minor adjustments in style and formatting.

Revision 1.0, IP Version 1.0, July 2017

Section	Change Summary
All	Initial release.

www.latticesemi.com