



MIPI D-PHY to CMOS Interface Bridge Soft IP

Supporting MIPI CSI-2 and MIPI DSI for Image Sensors and Displays

User Guide

FPGA-IPUG-02004-1.4

April 2019

Contents

1. Introduction	4
1.1. Quick Facts	5
1.2. Features.....	5
1.3. Conventions.....	5
1.3.1. Nomenclature.....	5
1.3.2. Data Ordering and Data Types	5
1.3.3. Signal Names	6
2. Functional Description.....	7
2.1. Top.....	7
2.2. D-PHY Common Interface Wrapper	10
2.3. Rx Global Operations Controller	10
2.4. Capture Controller.....	11
2.5. Byte2Pixel.....	12
2.6. Reset and Clocking	12
3. Parameter Settings	15
4. IP Generation and Evaluation	16
4.1. Licensing the IP.....	16
4.2. Getting Started	16
4.3. Generating IP in Clarity Designer	17
4.4. Generated IP Directory Structure and Files.....	20
4.5. Running Functional Simulation	22
4.6. Simulation Strategies	24
4.7. Simulation Environment.....	24
4.8. Instantiating the IP	25
4.9. Synthesizing and Implementing the IP	26
4.10. Hardware Evaluation.....	26
4.10.1. Enabling Hardware Evaluation in Diamond	26
4.11. Updating/Regenerating the IP.....	27
4.11.1. Regenerating an IP in Clarity Designer	27
References	28
Technical Support Assistance	28
Appendix A. Resource Utilization	29
Appendix B. What is Not Supported	30
Revision History	31

Figures

Figure 1.1. MIPI DSI to CMOS Interface Bridge IP System Diagram.....	4
Figure 1.2. MIPI CSI-2 to CMOS Interface Bridge IP System Diagram.....	4
Figure 2.1. MIPI D-PHY to CMOS Interface Bridge IP Block Diagram.....	7
Figure 2.2. MIPI DSI to CMOS Interface Bridge IP Block Diagram.....	8
Figure 2.3. MIPI CSI-2 to CMOS Interface Bridge IP Block Diagram.....	9
Figure 2.4. High-Speed Data Transmission	9
Figure 2.5. CMOS Transmit Interface Timing Diagram (DSI)	10
Figure 2.6. CMOS Transmit Interface Timing Diagram (CSI-2).....	10
Figure 2.7. MIPI D-PHY Clock Lane Module State Diagram	11
Figure 2.8. MIPI D-PHY Data Lane Module State Diagram	11
Figure 4.1. Clarity Designer Window	16
Figure 4.2. Starting Clarity Designer from Diamond Design Environment	17
Figure 4.3. Configuring MIPI D-PHY to CMOS Interface Bridge IP in Clarity Designer	18
Figure 4.4. Configuration Tab in IP User Interface.....	19
Figure 4.5. Video Tab in IP User Interface	20
Figure 4.6. MIPI D-PHY to CMOS Interface Bridge IP Directory Structure	20
Figure 4.7. Simulation Environment Block Diagram	24
Figure 4.8. DSI Model Video Data	25
Figure 4.9. CSI-2 Model Video Data	25
Figure 4.10. Regenerating IP in Clarity Designer	27

Tables

Table 1.1. MIPI D-PHY to CMOS Interface Bridge IP Quick Facts	5
Table 1.2. Pixel Data Order	6
Table 2.1. MIPI D-PHY to CMOS Interface Bridge IP Pin Function Description	7
Table 2.2. Capture Controller Outputs	12
Table 2.3. Clock Frequency Calculations.....	13
Table 2.4. Multiplier for Computing Reference Clock Frequency.....	13
Table 2.5. Supported Frequencies for MIPI D-PHY to CMOS Interface Bridge IP Configurations.....	14
Table 3.1. MIPI D-PHY to CMOS Interface Bridge IP Parameter Settings	15
Table 4.1. Files Generated by Clarity Designer	21
Table 4.2. Testbench Directives Common for DSI and CSI-2	22
Table 4.3. Testbench Directives for D-PHY Timing Parameters	23
Table 4.4. Testbench Directives for Reference Clock Period	23
Table 4.5. Testbench Directives for DSI Rx Type.....	23
Table 4.6. Testbench Directives for CSI-2 Rx Type.....	24
Table A.1. Resource Utilization.....	29

1. Introduction

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

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

A bridge is often required to connect a processor with a MIPI DSI interface to a display with an RGB interface or a camera with a MIPI CSI-2 interface to a processor with CMOS interface. The Lattice Semiconductor MIPI D-PHY to CMOS Interface Bridge IP provides this conversion for Lattice Semiconductor CrossLink™ devices. This is useful for wearable, tablet, human machine interfacing, medical equipment and many other applications.

This user guide is for MIPI D-PHY to CMOS Interface Bridge Soft IP design version 1.x.



Figure 1.1. MIPI DSI to CMOS Interface Bridge IP System Diagram



Figure 1.2. MIPI CSI-2 to CMOS Interface Bridge IP System Diagram

1.1. Quick Facts

Table 1.1. provides quick facts about the MIPI D-PHY to CMOS Interface Bridge IP for Lattice CrossLink device.

Table 1.1. MIPI D-PHY to CMOS Interface Bridge IP Quick Facts

		MIPI D-PHY to CMOS Interface Bridge IP Configuration	
		4-Lane MIPI DSI to RGB888 (HS_LP)	4-Lane MIPI CSI-2 to RGB888 (HS_LP)
Core Requirements	FPGA Families Supported	CrossLink	
	Targeted Device	LIF-MD6000-6MG81I	
Resource Utilization	Data Path Width	8 bits per lane, 32 bits total for 4 lanes	
	LUTs	819	815
	sysMEM™ EBRs	3	3
	Registers	792	764
	Programmable I/O	30	29
	Hard D-PHY	1	1
Design Tool Support	Lattice Implementation	Lattice Diamond® 3.8 or later	
	Synthesis	Lattice Synthesis Engine	
		Synopsys® Synplify Pro® L-2016.03L	
	Simulation	Aldec® Active-HDL™ 10.3 Lattice Edition	

1.2. Features

The key features of the MIPI D-PHY to CMOS Interface Bridge IP are:

- Compliant with MIPI D-PHY v1.1, MIPI DSI v1.1 and MIPI CSI-2 v1.1 specifications
- Supports MIPI D-PHY interfacing from 160 Mb/s up to 1.5 Gb/s
- Supports 1, 2, or 4 data lanes and one clock lane
- Supports continuous and non-continuous MIPI D-PHY clock
- Supports common MIPI DSI compatible video formats (RGB888, RGB666)
- Supports common MIPI CSI-2 compatible video formats (RGB888, RAW, YUV)
- Supports MIPI DSI Video Mode operation of Non-Burst Mode with Sync Pulses
- Supports dedicated End of Transmission short packet (EoTp)

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

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

Table 1.2 lists pixel data order coming from core module.

Table 1.2. Pixel Data Order

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

1.3.3. Signal Names

Signal names that end with:

- `_n` are active low
- `_i` are input signals
Some signals are declared as bidirectional (I/O) but are only used as input. Hence, `_i` identifier is used.
- `_o` are output signals
Some signals are declared as bidirectional (I/O) but are only used as output. Hence, `_o` identifier is used.
- `_io` are bidirectional signals

2. Functional Description

The MIPI D-PHY to CMOS Interface Bridge IP serves as a bridge between a MIPI DSI host and a display device or MIPI CSI-2 host with applications processor.

2.1. Top

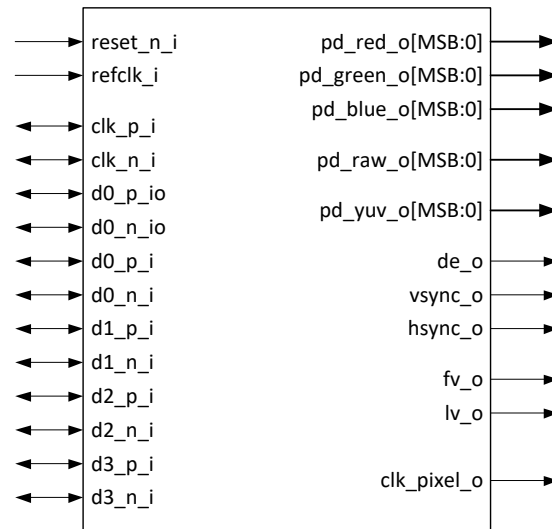


Figure 2.1. MIPI D-PHY to CMOS Interface Bridge IP Block Diagram

Table 2.1. MIPI D-PHY to CMOS Interface Bridge IP Pin Function Description

Port Name	Direction	Function description
Clock and Reset		
refclk_i	I	Reference clock for internal PLL. Available only when MIPI D-PHY clock is non-continuous.
reset_n_i	I	Asynchronous system reset (active low)
MIPI Interface		
clk_p_i, clk_n_i	I/O	MIPI D-PHY clock lane
d0_p_io, d0_n_io	I/O	MIPI D-PHY data lane 0. Available only for MIPI DSI configuration.
d0_p_i, d0_n_i	I/O	MIPI D-PHY data lane 0. Available only for MIPI CSI-2 configuration.
d1_p_i, d1_n_i	I/O	MIPI D-PHY data lane 1. Available only for configurations with two or more data lanes.
d2_p_i, d2_n_i	I/O	MIPI D-PHY data lane 2. Available only for configurations with four data lanes
d3_p_i, d3_n_i	I/O	MIPI D-PHY data lane 3. Available only for configurations with four data lanes.
CMOS Interface		
clk_pixel_o	O	Pixel clock from internal PLL
vsync_o	O	Vertical Sync Indicator (active high). Goes high when VSYNC start short packet is received. Goes low when VSYNC end short packet is received. Available only for MIPI DSI mode.
hsync_o	O	Horizontal Sync Indicator (active high). Goes high when either VSYNC/HSYNC start or VSYNC end short packet is received. Goes low when HSYNC end short packet is received. Available only for MIPI DSI mode.
de_o	O	Data Enable Indicator (active high). Goes high at the start of valid data long packet and goes low at the end of valid data long packet. Available only for MIPI DSI mode.

Port Name	Direction	Function description
fv_o	O	Frame Valid Indicator (active high). Goes high when frame start short packet is received and goes low when frame end short packet is received. Available only for MIPI CSI-2 mode.
lv_o	O	Line Valid Indicator (active high). Goes high at the start of valid data long packet and goes low at the end of valid data long packet. Available only for MIPI CSI-2 mode.
pd_red_o [MSB:0]	O	Red component of pixel data. Data width depends on selected data type. Available only when any RGB data type is selected.
pd_green_o [MSB:0]	O	Green component of pixel data. Data width depends on selected data type. Available only when any RGB data type is selected.
pd_blue_o [MSB:0]	O	Blue component of pixel data. Data width depends on selected data type. Available only when any RGB data type is selected.
pd_raw_o [MSB:0]	O	Raw pixel data. Data width depends on selected data type. Available only when any RAW data type is selected.
pd_yuv_o [MSB:0]	O	YUV pixel data. Data width depends on selected data type. Available only when any YUV data type is selected.

Figure 2.2 shows the MIPI DSI to CMOS Interface Bridge IP block diagram.

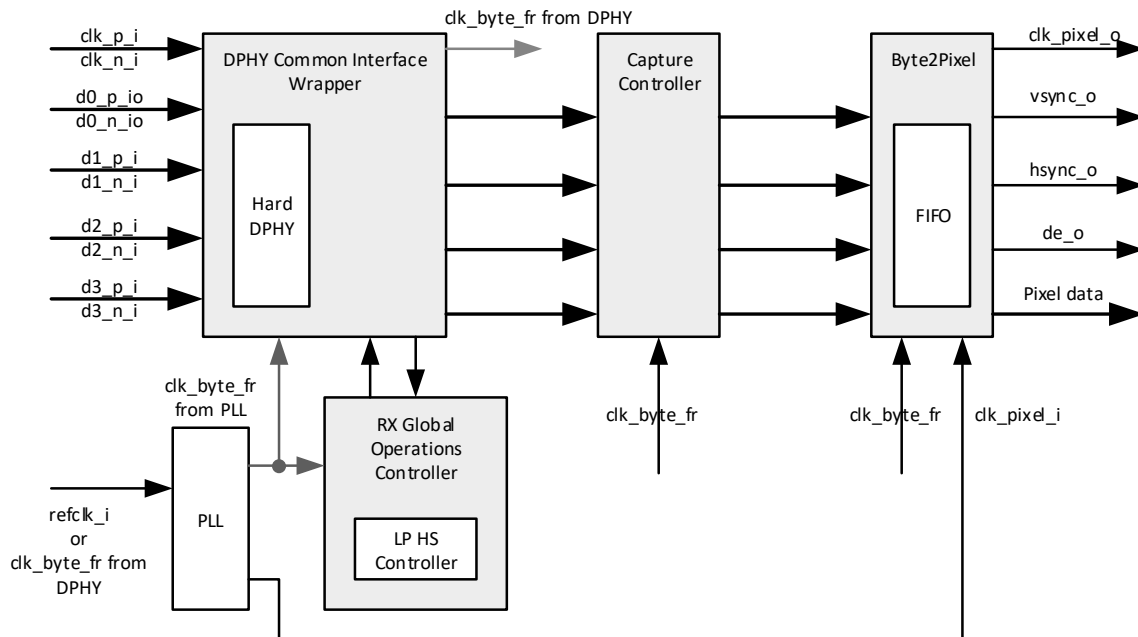


Figure 2.2. MIPI DSI to CMOS Interface Bridge IP Block Diagram

Figure 2.3 shows the MIPI CSI-2 to CMOS Interface Bridge IP block diagram.

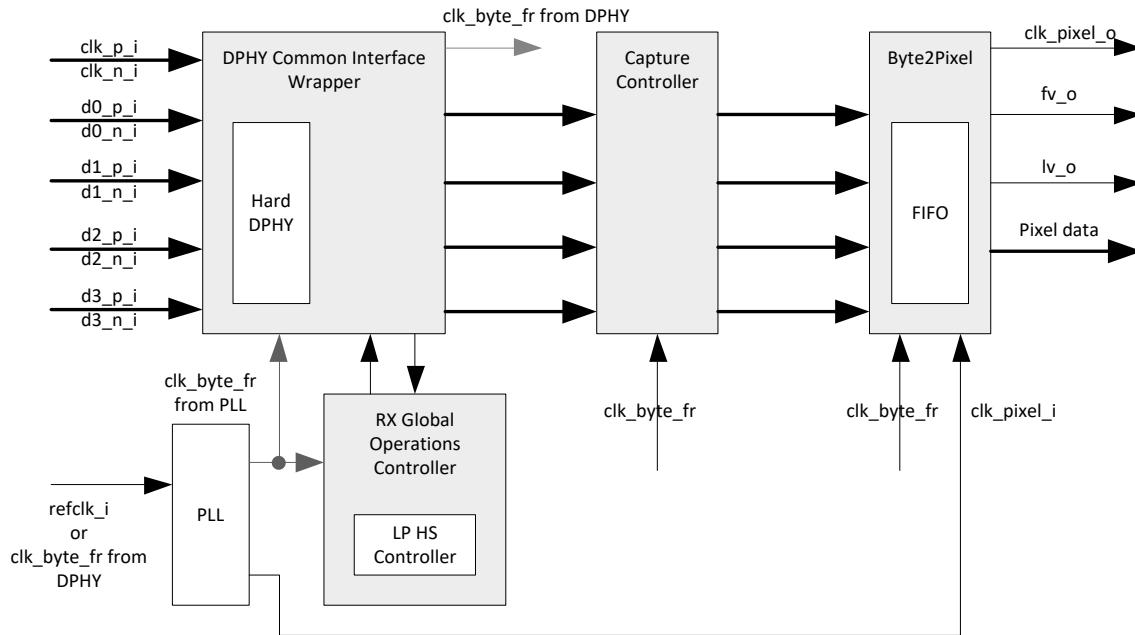


Figure 2.3. MIPI CSI-2 to CMOS Interface Bridge IP Block Diagram

The MIPI D-PHY receive interface has one clock lane and configurable number of data lanes. The clock lane is center-aligned to the data lanes. The MIPI D-PHY clock can either be continuous (high speed only) or non-continuous. When the MIPI D-PHY clock is non-continuous, proper transition from low power (LP) to high speed (HS) mode of clock lane is required. The data lanes also require proper transition from LP to HS modes. In HS mode, data stream from each data lane is deserialized to byte data. The deserialization is done with 1:8 gearing. The byte data is word-aligned based on the SoT Sync sequence defined in the MIPI D-PHY Specification version 1.1.

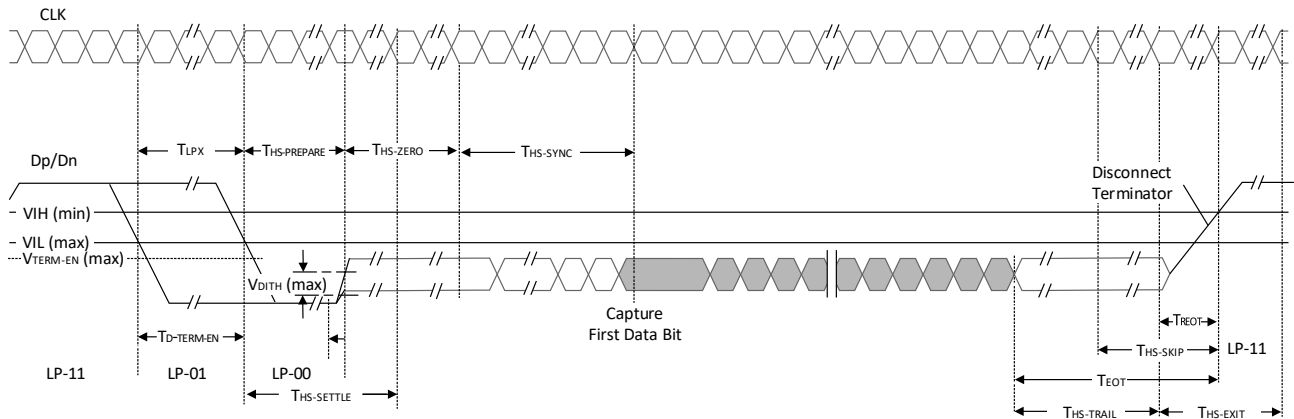


Figure 2.4. High-Speed Data Transmission

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

The clock is center-aligned against data and control signals. All signal transitions happen in sync with the falling edge of pixel clock as shown in Figure 2.5 and Figure 2.6.

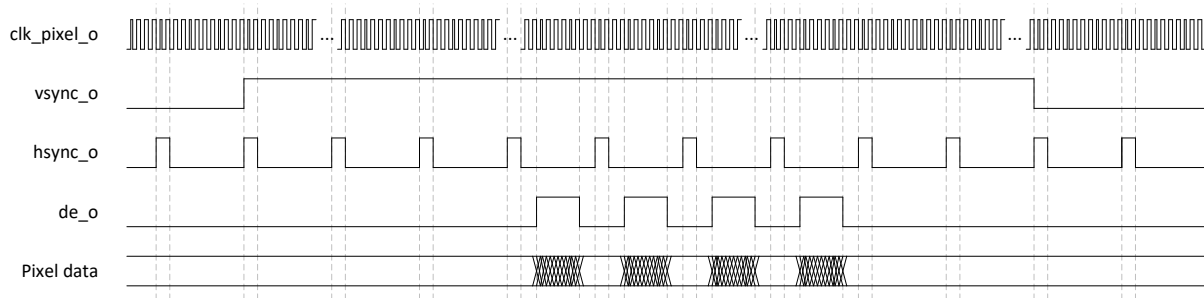


Figure 2.5. CMOS Transmit Interface Timing Diagram (DSI)

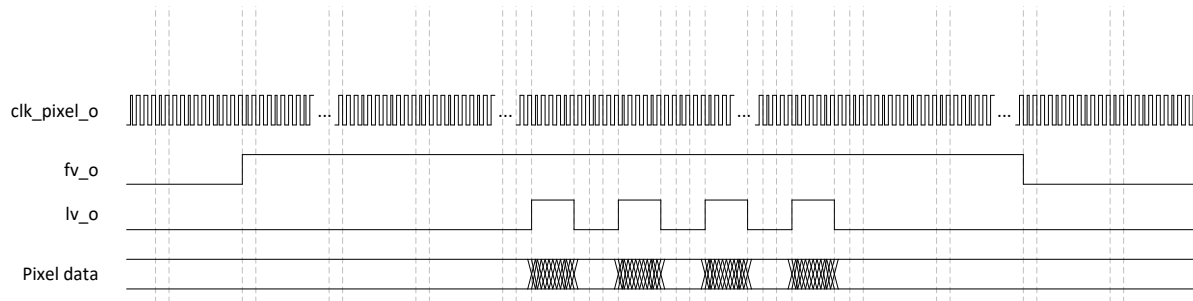


Figure 2.6. CMOS Transmit Interface Timing Diagram (CSI-2)

2.2. D-PHY Common Interface Wrapper

This block instantiates and configures hard D-PHY IP to receive MIPI D-PHY high-speed data from all enabled data lanes. The hard D-PHY IP outputs 8-bit parallel data in non-continuous byte clock domain for each data lane.

Byte data are transferred to continuous byte clock domain using multicycle registers. Data enable signal from this block becomes active when SoT Sync is successfully detected by hard D-PHY IP from all enabled data lanes and becomes inactive when MIPI D-PHY data lanes go to Stop state (LP11).

2.3. Rx Global Operations Controller

This block controls the high-speed termination enable of MIPI D-PHY clock and data lanes. When MIPI D-PHY clock is continuous, the HS termination enable of clock lane is tied to VCC. When MIPI D-PHY clock is non-continuous, the HS termination enable of clock lane becomes active after proper LP to HS transition is observed. A reference clock input is required for this function. The required LP to HS transition on clock lane is shown in [Figure 2.7](#) as per MIPI D-PHY Specification version 1.1.

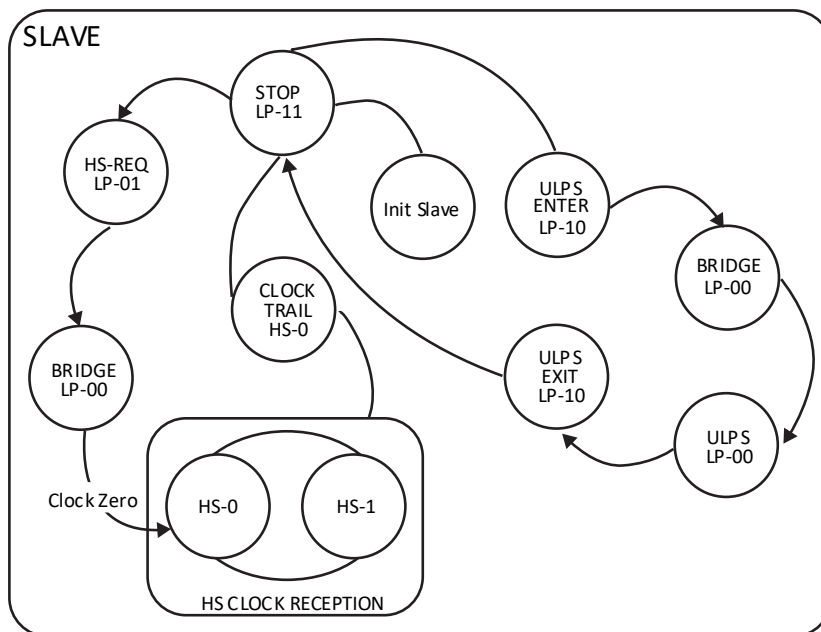


Figure 2.7. MIPI D-PHY Clock Lane Module State Diagram

Similarly, HS termination enable of data lanes becomes high after proper LP to HS transition is detected on data lane 0. A free-running byte clock is used for this function. The required LP to HS transition on data lanes is shown in [Figure 2.8](#) as per MIPI D-PHY Specification version 1.1.

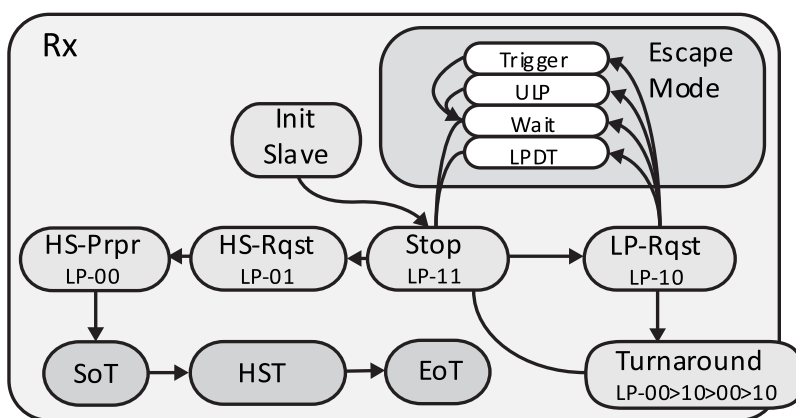


Figure 2.8. MIPI D-PHY Data Lane Module State Diagram

2.4. Capture Controller

This block takes data bytes from D-PHY Common Interface Wrapper and detects short and long packets defined by MIPI DSI or MIPI CSI-2 to generate sync signals and extract video data and other control parameters. Optional line start and line end short packets for MIPI CSI-2 are also detected but is not translated into a line valid signal. This block has outputs that are not applicable to MIPI D-PHY to CMOS Interface Bridge IP. [Table 2.2](#) lists all outputs relevant to MIPI D-PHY to CMOS Interface Bridge IP.

Table 2.2. Capture Controller Outputs

Port Name	Direction	Function Description
bd_o[MSB:0]	Output	Registered data bytes. Data bytes are delayed by a few byte clock cycles to be in sync with extracted video data and control outputs. Data width depends on number of active MIPI D-PHY data lanes
payload_en_o	Output	Payload data enable to indicate when byte to pixel conversion is required (active high)
payload_o[MSB:0]	Output	Video data or payload. Data width depends on number of active MIPI D-PHY data lanes
sp_en_o	Output	Short packet enable. Goes high for 1 byte clock cycle when short packet is detected (active high)
lp_en_o	Output	Long packet enable. Goes high for 1 byte clock cycle when long packet is detected (active high)
lp_av_en_o	Output	Long packet enable for active video data. Goes high for 1 byte clock cycle when long packet containing active video is detected (active high)
vc_o[1:0]	Output	Virtual channel
wc_o[15:0]	Output	Word count of long packet
dt_o[5:0]	Output	Data type
ecc_o[7:0]	Output	ECC of packet header

2.5. Byte2Pixel

This block converts byte data into pixel data using FIFO. Continuous byte clock is used to write data to FIFO while pixel clock is used to read data from FIFO.

The VSYNC and HSYNC outputs are also generated by this block and transferred to pixel clock domain using synchronization registers. Since only DSI Non-Burst Mode with Sync Pulses is supported, the generation of VSYNC and HSYNC control signals is dependent on the MIPI DSI host device as follows. VSYNC goes active high and inactive low when the *VSYNC Start* and *VSYNC End* short packets are seen, respectively. HSYNC goes active high when the *HSYNC Start*, *VSYNC Start*, and *VSYNC End* short packets are seen. HSYNC goes inactive low when the *HSYNC end* short packet is seen. MIPI DSI Non-Burst Mode with Sync Events and Burst Mode operations are not supported.

2.6. Reset and Clocking

Asynchronous active low reset input (reset_n_i) is used as a system reset. To reset logic in continuous byte clock domain, system reset passes through synchronization registers to create asynchronous reset assertion and synchronous reset deassertion. The same thing is done to reset logic in pixel clock domain, but with additional condition that internal PLL lock must be high. The system reset input must be asserted for at least three times the slowest clock (byte clock or pixel clock).

Internal PLL could take ~15 ms to be locked after PLL reference clock is made available. Data loss is expected when incoming MIPI D-PHY transaction begins during this period when PLL lock is not yet obtained. To avoid malfunction, the MIPI D-PHY to CMOS Interface Bridge IP discards any received MIPI DSI or MIPI CSI-2 packets until it detects *VSYNC start* short packet in case of MIPI DSI or *Frame end* short packet and HS to LP to HS transition in case of MIPI CSI-2.

When MIPI D-PHY clock is continuous, it is expected to be in high speed mode at power on of the device. The HS termination enable of clock lane is tied to VCC. Continuous byte clock is generated by hard D-PHY IP and used as PLL reference clock. Internal PLL generates pixel clock. A special case is 1-lane with any 8-bit RAW or YUV data type where internal PLL is disabled since continuous byte clock is also used as pixel clock.

When MIPI D-PHY clock is non-continuous, a reference clock input is required for detecting LP to HS transition of clock lane and PLL reference clock. Internal PLL generates byte clock and pixel clock.

Frequency calculations are given in [Table 2.3](#). DCK refers to MIPI D-PHY clock frequency.

Table 2.3. Clock Frequency Calculations

Clock	Formula
Rx line rate	DCK * 2
Tx line rate	Pixel clock
D-PHY clock	DCK
Byte clock	DCK / (Rx gear / 2)
Pixel clock (Output)	Byte clock * Rx lanes * Rx gear / Pixel width
Reference clock	DCK / N , where N is defined in Table 2.4

Table 2.4. Multiplier for Computing Reference Clock Frequency

Data Type	Rx Lanes	Rx Line Rate (Mb/s)	N
RGB666	4	160 – 1500	8
	2	160 – 900	12
		> 900	16
	1	160 – 900	12
		> 900	24
RGB888/ RGB666_LOOSE	4/2/1	160 – 900	12
		> 900	24
RAW8/ YUV 8bit	4	160 – 600	4
	2	160 – 900	8
		> 900	16
	1	160 – 900	12
		> 900	16
RAW10/ YUV 10bit	4	160 – 750	5
	2	160 – 1500	10
	1	160 – 900	10
		> 900	20
RAW12/ YUV 12bit	4	160 – 450	6
		> 450	12
	2/1	160 – 900	12
		> 900	24

Table 2.5. Supported Frequencies for MIPI D-PHY to CMOS Interface Bridge IP Configurations

Data Type	Rx Lanes	Min DCK	Max DCK	Byte Clock*	Pixel Clock*	Refclk*
RGB666	4	80	675	84.375	300	84.375
	2	80	750	93.75	166.67	46.875
	1	80	750	93.75	83.33	31.25
RGB888/ RGB666_LOOSE	4	80	750	93.75	250	31.25
	2	80	750	93.75	125	31.25
	1	80	750	93.75	62.5	31.25
RAW8/ YUV 8bit	4	80	300	75	300	75
	2	80	600	75	300	37.5
	1	80	750	93.75	187.5	46.875
RAW10/ YUV 10bit	4	80	375	93.75	300	75
	2	80	750	93.75	300	75
	1	80	750	93.75	150	37.5
RAW12/ YUV 12bit	4	80	450	112.5	300	37.5
	2	80	750	93.75	250	31.25
	1	80	750	93.75	125	31.25

***Note:** Clock frequencies are computed based on maximum DCK.

3. Parameter Settings

Table 3.1 lists the parameters used to generate MIPI D-PHY to CMOS Interface Bridge IP.

Table 3.1. MIPI D-PHY to CMOS Interface Bridge IP Parameter Settings

Parameter	Attribute	Options	Description
Number of Rx Channels	Fixed	1	Number of MIPI D-PHY channels
Rx Interface	User-Input	MIPI DSI or MIPI CSI-2	Receive interface
Number of Rx Lanes	User-Input	1, 2, or 4	Number of MIPI D-PHY data lanes
Rx Gearing	Read-Only	8, 16	Rx gear is automatically computed based on selected configuration and D-PHY data rate
Rx D-PHY IP	Fixed	Hard D-PHY	MIPI D-PHY Implementation
Number of Tx Channels	Fixed	1	Number of CMOS interface
Tx Interface	Fixed	CMOS	Transmit interface
Number of Tx Lanes	Read-Only	8, 10, 12, 18 or 24	Pixel width derived from data type
Tx Gearing	Read-Only	1, 2	Tx gear is automatically computed based on selected configuration and D-PHY data rate
Rx Line Rate	User-Input	See Table 2.5	Data rate per MIPI D-PHY data lane
Tx Line Rate	Read-Only	See Table 2.3	Data rate per CMOS lane
D-PHY Clock Frequency	Read-Only	See Table 2.3	MIPI D-PHY clock frequency (DCK). $t_{HS-SETTLE}$ MIPI D-PHY timing parameter is also derived from this setting (85 ns + 6 UI). $t_{HS-SETTLE}$ counter is implemented in byte clock domain. The expected actual $t_{HS-SETTLE}$ is ~2 byte clock cycles more than the computed value.
D-PHY Clock Mode	User-Input	Continuous or Non-continuous	MIPI D-PHY clock mode: <ul style="list-style-type: none"> Continuous (HS_ONLY) Non-continuous (HS_LP)
Byte Clock Frequency	Read-Only	See Table 2.3	Byte clock frequency
Pixel Clock Frequency	Read-Only	See Table 2.3	Pixel clock frequency
Reference Clock Frequency	Read-Only	See Table 2.3	Reference clock frequency
Data Type	User-Input	RGB888 RGB666 RAW8 RAW10 RAW12 YUV420_8 YUV420_8_CSPS LEGACY_YUV420_8 YUV422_8 YUV420_10 YUV420_10_CSPS YUV422_10	Supported MIPI DSI data types: RGB888, RGB666 Supported MIPI CSI-2 data types: RGB888, RAW8, RAW10, RAW12, YUV420_8, YUV420_8_CSPS, LEGACY_YUV420_8, YUV422_8, YUV420_10, YUV420_10_CSPS, YUV422_10
RGB666 Type	User-Input	Packed or Loosely Packed	Selects between RGB666 Packed and RGB666 Loosely Packed MIPI DSI formats

4. IP Generation and Evaluation

This section provides information on how to generate Lattice MIPI D-PHY to CMOS Interface Bridge IP using the Diamond Clarity Designer, and how to run simulation, synthesis and hardware evaluation.

4.1. Licensing the IP

An IP-specific license is required to enable full, unrestricted use of the MIPI D-PHY to CMOS Interface Bridge IP in a complete, top-level design. The MIPI D-PHY to CMOS Interface Bridge IP is available free of charge.

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 IP for your design.

You may download or generate the MIPI D-PHY to CMOS Interface Bridge IP and fully evaluate it through functional simulation and implementation (synthesis, map, place and route) without the IP license. The MIPI D-PHY to CMOS Interface Bridge IP also supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP that operate in hardware for a limited time (approximately four hours) without requiring an IP license. 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 MIPI D-PHY to CMOS Interface Bridge 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 has been installed, the IP is available in the Clarity Designer user interface as shown in [Figure 4.1](#).

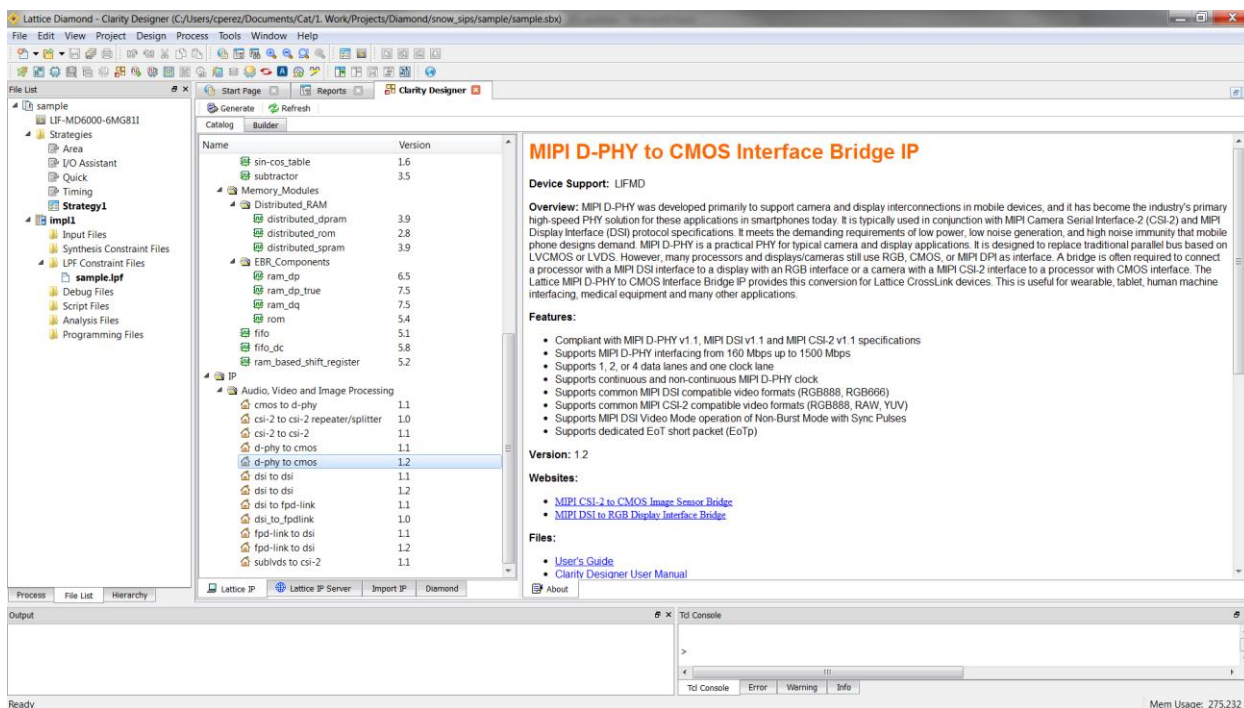



Figure 4.1. Clarity Designer Window

4.3. Generating IP in Clarity Designer

The Clarity Designer tool is used to customize modules and IPs and place them into the device's architecture. The following describes the procedure for generating MIPI D-PHY to CMOS Interface Bridge IP in Clarity Designer.

Clarity Designer is started from the Diamond design environment.

To start Clarity Designer:

1. Create a new empty Diamond project for CrossLink 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 fill out the following items as shown in Figure 4.2.
 - **Create new Clarity design** – Choose to create a new Clarity Design project directory in which the MIPI D-PHY to CMOS Interface Bridge IP is generated.
 - **Design Location** – Clarity Design project directory path.
 - **Design Name** – Clarity Design project name.
 - **HDL Output** – Hardware Description Language Output Format (Verilog).

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.

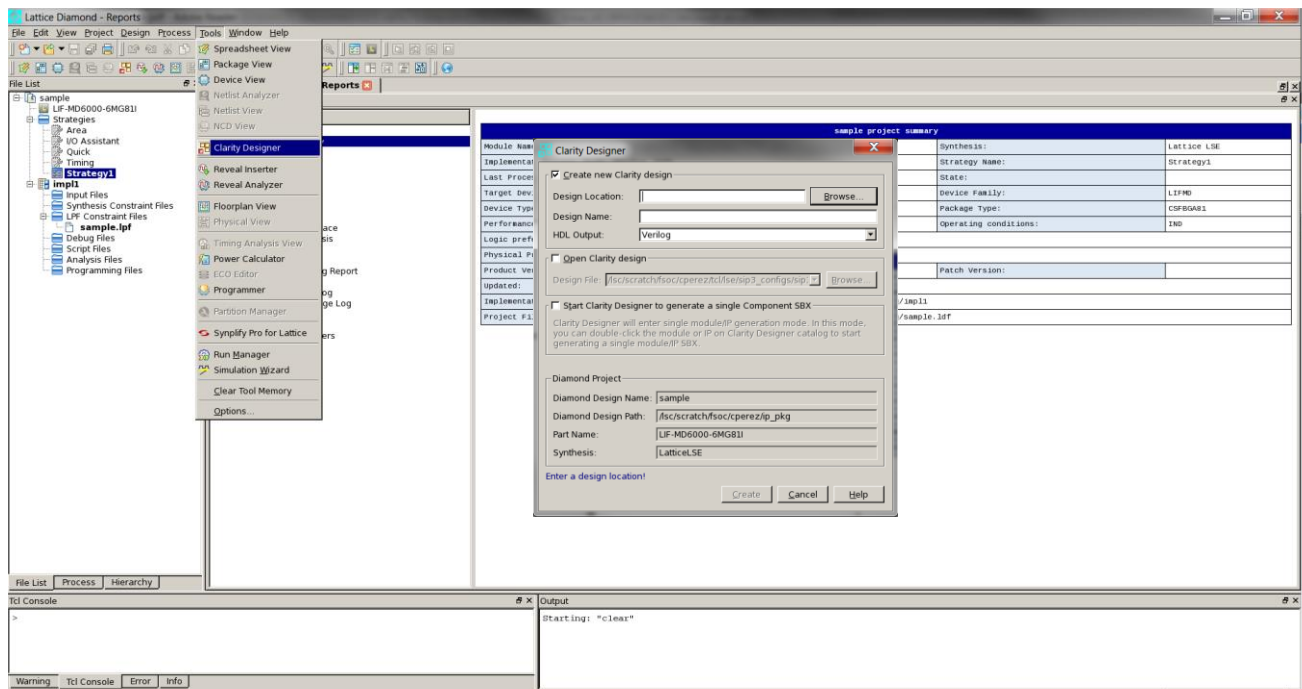


Figure 4.2. Starting Clarity Designer from Diamond Design Environment

To configure the MIPI D-PHY to CMOS Interface Bridge IP in Clarity Designer:

1. Double-click **d-phy to cmos** in the IP list of the Catalog view. The **d-phy to cmos** dialog box is displayed as shown in Figure 4.3.

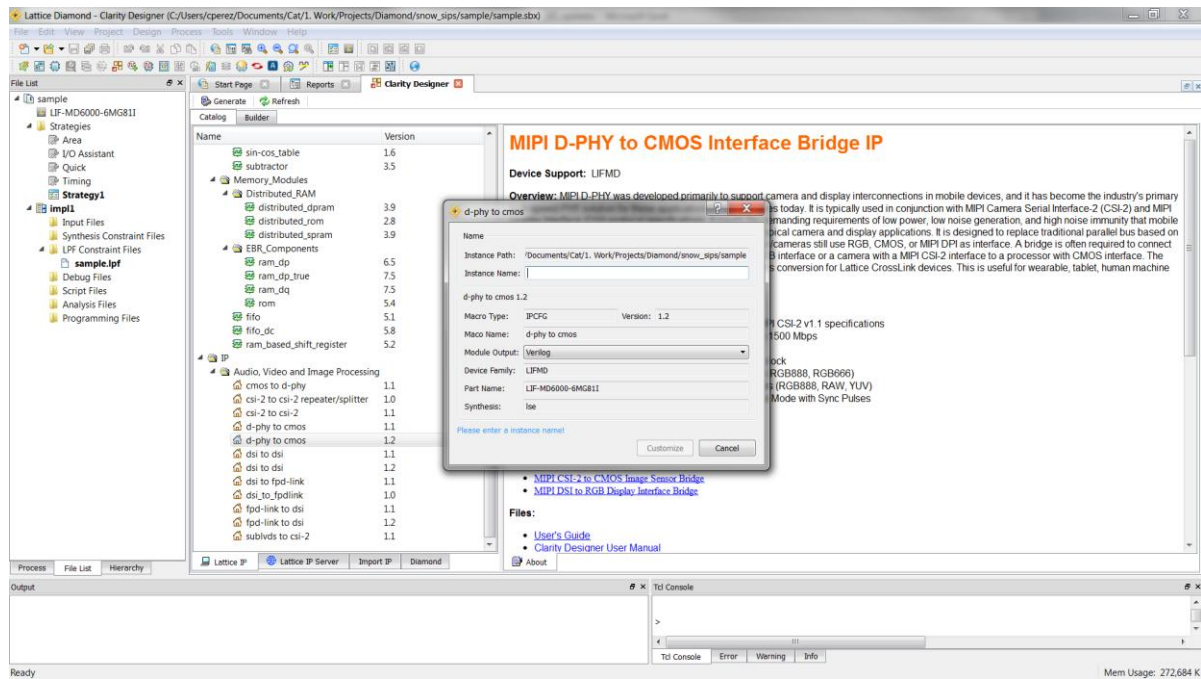


Figure 4.3. Configuring MIPI D-PHY to CMOS Interface Bridge IP in Clarity Designer

2. Enter the **Instance Name**.
3. Click the **Customize** button. An IP configuration interface is displayed as shown in Figure 4.4 and Figure 4.5. From this dialog box, you can select the IP parameter options specific to your application. The parameters are grouped into two tabs: **Configuration** and **Video**.

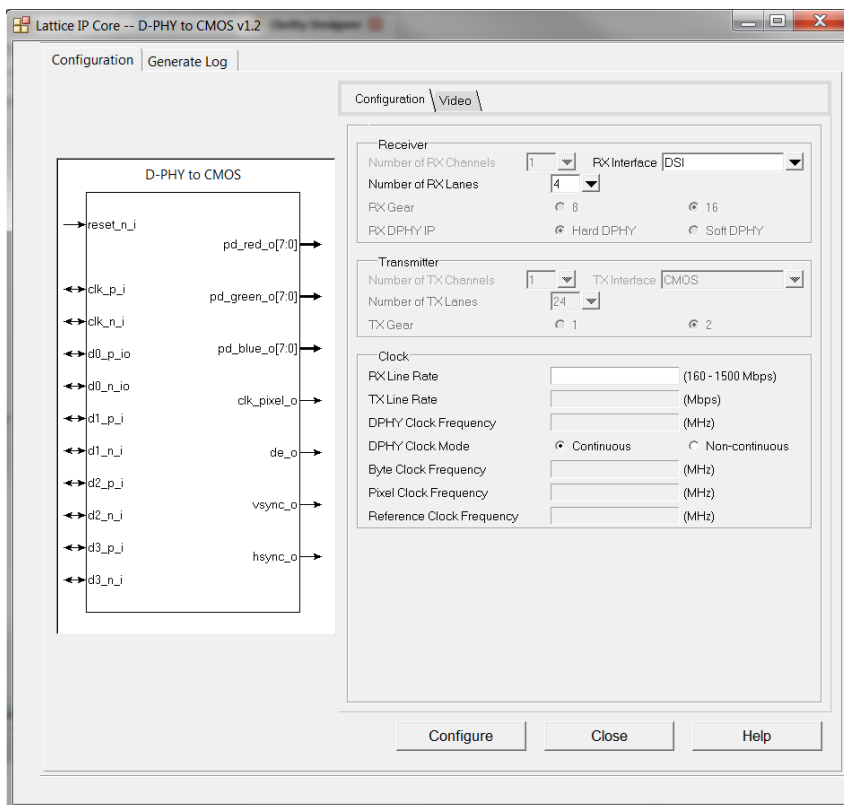


Figure 4.4. Configuration Tab in IP User Interface

4. Select the required parameters, and click the **Configure** button.
5. Click **Close**.

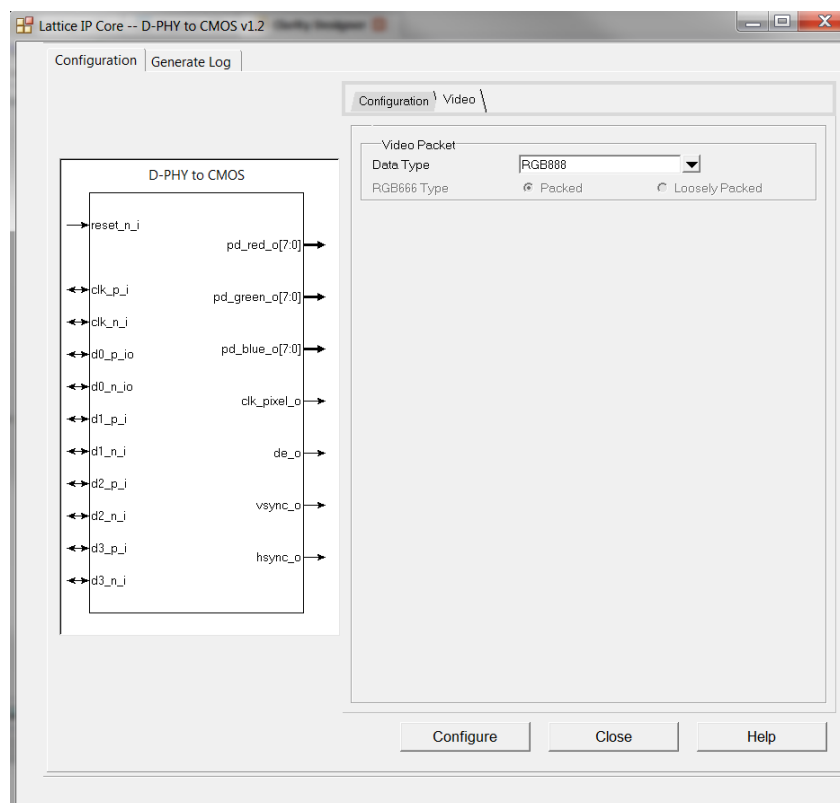



Figure 4.5. Video Tab in IP User Interface

- 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.

4.4. Generated IP Directory Structure and Files

The directory structure of the generated IP files is shown in Figure 4.6.

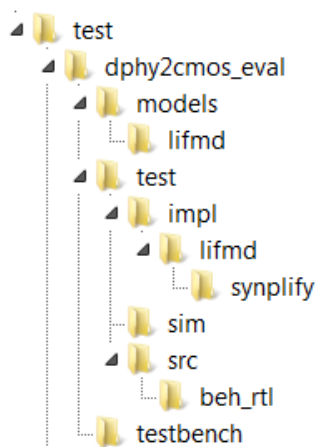


Figure 4.6. MIPI D-PHY to CMOS Interface Bridge IP Directory Structure

The design flow for the IP created with Clarity Designer uses post-synthesized modules (NGO) of the IP core modules for synthesis and uses protected models for simulation. The post-synthesized modules are customized when you configure the IP and created automatically when the IP is generated. The protected models are common to all configurations.

Table 4.1 provides a list of key files and directories created by Clarity Designer with details on how they are used.

Table 4.1. Files Generated by Clarity Designer

File	Description
<instance_name>.v	Verilog top-level module of MIPI D-PHY to CMOS Interface Bridge 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 which 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 the 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>_inst.v/vhd	Template for instantiating the generated soft IP top-level in another user-created top module.

All IP files are generated inside \<project_dir> directory (test in Figure 4.6). The \<project_dir> is <Design Location>\<Design Name>\<Instance Name>, see the [Generating IP in Clarity Designer](#) section. A separate \<project_dir> is created each time MIPI D-PHY to CMOS Interface Bridge IP is created with a different IP instance name.

The \dphy2cmos_eval and subdirectories provide files supporting push-button IP evaluation through functional simulations, design implementation (synthesis, map, place and route) and hardware evaluation. Inside the \dphy2cmos_eval is the \<instance_name> folder (test in Figure 4.6) which contains protected behavioral files in \<instance_name>\src\beh_rtl and a pre-built Diamond project in \<instance_name>\impl\lifmd\<synthesis_tool>. The \<instance_name> is the IP instance name specified by user in Clarity Designer. The simulation part of user evaluation provides testbench and test cases supporting RTL simulation for Active-HDL simulator under \testbench folder. Separate directories located at \<project_dir>\dphy2cmos_eval\<instance_name>\sim\aldec are provided and contain specific pre-built simulation script files. See the [Running Functional Simulation](#) section below for details.

The pll_wrapper model in \<project_dir>\models\lifmd is used for both simulation and implementation.

4.5. Running Functional Simulation

The generated IP package includes the behavioral models (<instance_name>_*_beh.v) provided in <project_dir>\dphy2cmos_eval\<instance_name>\src\beh_rtl for functional simulation. PLL wrapper (pll_wrapper.v) in <project_dir>\dphy2cmos_eval\models\liffmd and parameters file (<instance_name>_params.v) in <project_dir> are also needed for functional simulation. The testbench files are provided in <project_dir>\dphy2cmos_eval\testbench.

To run the evaluation simulation on Active-HDL (Windows only):

1. Create new project using Lattice Diamond for Windows.
2. Open **Active-HDL** Lattice Edition user interface tool.
7. Modify the *.do file located in <project_dir>\dphy2cmos_eval\<instance_name>\sim\aldec\ .
 - a. Specify working directory. For example:
set sim_working_folder **C:/my_design**.
 - b. Specify workspace name that is created in working directory. For example:
set workspace_name **design_space**.
 - c. Specify design name, for example:
set design_name **DesignA**.
 - d. Specify design path where the IP Core generated using Clarity Designer is located. For example:
set design_path **C:/my_designs/DesignA**.
 - e. Specify design instance name (same as the instance name specified in Clarity Designer). For example:
set design_inst **DesignA_inst**.
 - f. Specify Lattice Diamond primitive path to where it is installed. For example:
set diamond_dir **C:/Iscc/diamond/3.8_x64**.
4. Update testbench parameters and/or directives to customize data size, clock and/or other settings. See [Table 4.2](#) for the list of valid testbench compiler directives.
5. Click **Tools > Execute macro**.
6. Select the *.do file.
7. Wait for the simulation to finish.

Testbench parameters and directives can be modified by setting the define in the vlog command in the *.do file.

[Table 4.2](#) is a list of testbench directives common for DSI and CSI-2 Rx type.

Table 4.2. Testbench Directives Common for DSI and CSI-2

Directive	Description
NUM_FRAMES	Used to set the number of video frames
NUM_LINES	Used to set the number of lines per frame
VIRTUAL_CHANNEL	Used to set the virtual channel number
DPHY_DEBUG_ON	Used to enable or disable debug messages 0 – Debug messages disabled 1 – Debug messages enabled
DPHY_CLK	Used to set the D-PHY clock period (in ps)
FRAME_LPM_DELAY	Used to set the low power mode delay between frames (in ps)

The testbench has default settings for D-PHY timing parameters. Refer to Table 14 of MIPI D-PHY Specification version 1.1 for information regarding D-PHY timing requirements. To modify the D-PHY timing parameters, you can set the following testbench directives:

Table 4.3. Testbench Directives for D-PHY Timing Parameters

Directive	Description
DPHY_LPX	Used to set T-LPX (in ps)
DPHY_CLK_PREPARE	Used to set T-CLK-PREPARE (in ps)
DPHY_CLK_ZERO	Used to set T-CLK-ZERO (in ps)
DPHY_CLK_PRE	Used to set T-CLK-PRE (in ps)
DPHY_CLK_POST	Used to set T-CLK-POST (in ps)
DPHY_CLK_TRAIL	Used to set T-CLK-TRAIL (in ps)
DPHY_HS_PREPARE	Used to set T-HS-PREPARE (in ps)
DPHY_HS_ZERO	Used to set T-HS-ZERO (in ps)
DPHY_HS_TRAIL	Used to set T-HS-TRAIL (in ps)

By default, the testbench automatically calculates the reference clock period for HS_LP clock mode. You can override the clock period by defining the following testbench directive:

Table 4.4. Testbench Directives for Reference Clock Period

Directive	Description
REF_CLK	Used to set the Reference clock period input to the design (in ps)

Table 4.5 is a list of testbench directives for DSI Rx type.

Table 4.5. Testbench Directives for DSI Rx Type

File	Description
LP_BLANKING	Used to drive low-power blanking. If this is not defined, the testbench drives HS data as blanking
DSI_VACT_PAYLOAD	Number of bytes of active pixels per line
DSI_HSA_PAYLOAD	Number of bytes of Horizontal Sync Active Payload (used for Non-burst sync pulse)
DSI_BLLP_PAYLOAD	Number of bytes of BLLP Payload (used for HS data blanking)
DSI_HBP_PAYLOAD	Number of bytes of Horizontal Back Porch Payload (used for HS data blanking, and in LP blanking for Non-burst sync pulse mode)
DSI_HFP_PAYLOAD	Number of bytes of Horizontal Front Porch Payload (used for HS data blanking, and in LP blanking for Non-burst sync pulse mode)
DSI_VSA_LINES	Number of Vertical Sync Active Lines
DSI_VBP_LINES	Number of Vertical Back Porch Lines
DSI_VFP_LINES	Number of Vertical Front Porch Lines
DSI_EOTP_ENABLE	Used to enable/disable transmission of EOTP packet 0 – EOTP packet is disabled 1 – EOTP packet is enabled
DSI_LPS_BLLP_DURATION	Used to set the duration (in ps) for BLLP low-power state (used for LP blanking)
DSI_LPS_HBP_DURATION	Used to set the duration (in ps) for Horizontal Back Porch low-power state (used for LP blanking in Non-burst sync events and Burst mode)
DSI_LPS_HFP_DURATION	Used to set the duration (in ps) for Horizontal Front Porch low-power state (used for LP blanking in Non-burst sync events and Burst mode)
NON_BURST_SYNC_EVENTS	Used to set the video mode type to Non-burst sync events (Not supported by DUT)
BURST_MODE	Used to set the video mode type to Burst Mode (Not supported by DUT)
NON_BURST_SYNC_PULSE	Used to set the video mode type to Non-burst sync pulse

Table 4.6 is a list of testbench directives for CSI-2 Rx type.

Table 4.6. Testbench Directives for CSI-2 Rx Type

File	Description
CSI2_LPS_GAP	Used to set low power state delay between HS transactions (in ps)
CSI2_NUM_PIXELS	Used to set the number of pixels per line
CSI2_LS_LE_EN	Used to enable/disable D-PHY model transmission of line start and line end packets 0 – No Line start and Line end packets 1 – Line start and Line end packets enable

4.6. Simulation Strategies

This section describes the simulation environment which demonstrates basic MIPI D-PHY to CMOS functionality.

Figure 4.7 shows the block diagram of simulation environment.

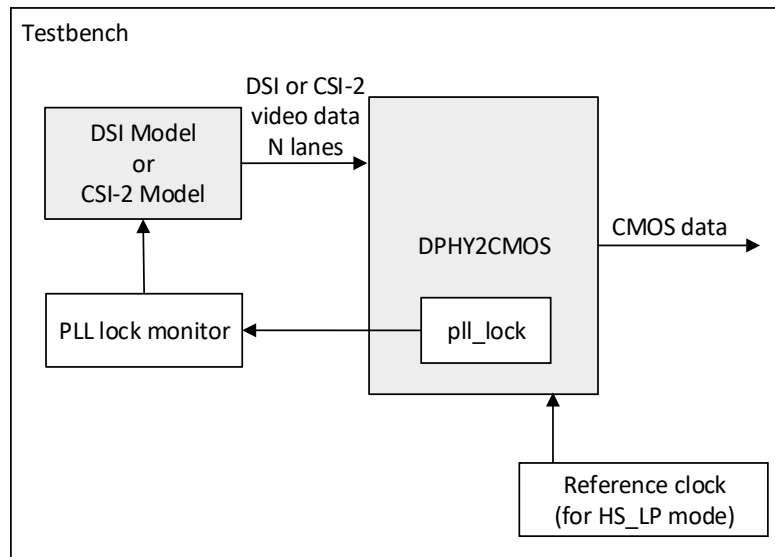


Figure 4.7. Simulation Environment Block Diagram

4.7. Simulation Environment

The simulation environment is made up of the DSI model instance if Rx type is DSI, or made up of CSI-2 model if Rx type is CSI-2. The instantiated model is connected to the MIPI D-PHY to CMOS Interface Bridge IP instance (DUT) in the testbench. The DSI model or CSI-2 model is configured based on the DUT configurations and testbench configurations. The testbench monitors assertion of the internal pll_lock signal before sending the video data to the DUT. The testbench also transmits reference clock to the DUT if D-PHY clock mode is non-continuous.

The video data transmitted by the DSI model can viewed in the waveform, see Figure 4.8:

- tb.dsi_ch0.data0 – refers to the data bytes transmitted in D-PHY data lane 0
- tb.dsi_ch0.data1 – refers to the data bytes transmitted in D-PHY data lane 1
- tb.dsi_ch0.data2 – refers to the data bytes transmitted in D-PHY data lane 2
- tb.dsi_ch0.data3 – refers to the data bytes transmitted in D-PHY data lane 3

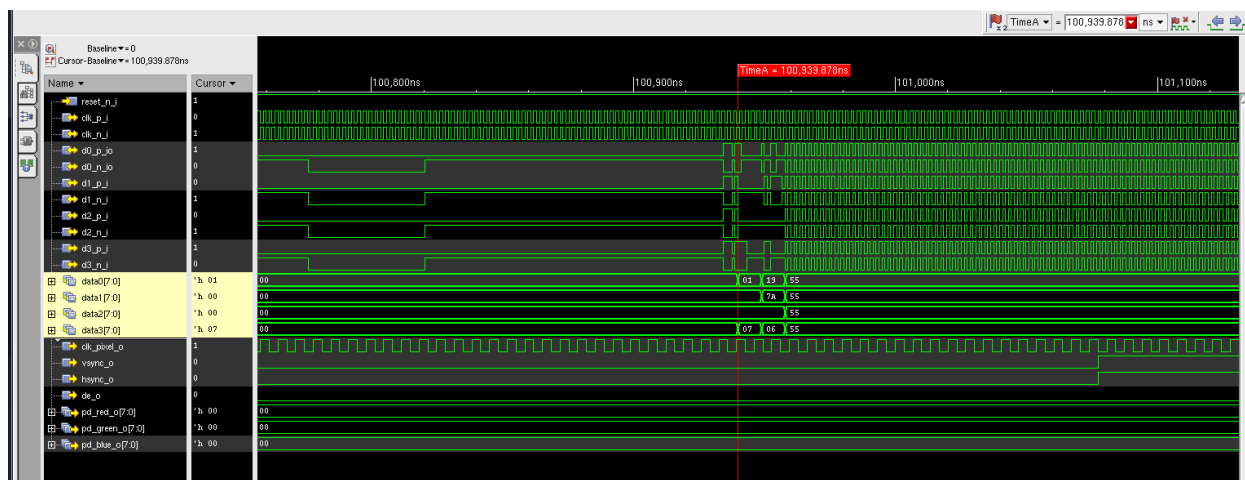


Figure 4.8. DSI Model Video Data

The video data transmitted by the CSI-2 model can viewed in the waveform, see Figure 4.9:

- tb.csi2_ch0.data0 – refers to the data bytes transmitted in D-PHY data lane 0
- tb.csi2_ch0.data1 – refers to the data bytes transmitted in D-PHY data lane 1
- tb.csi2_ch0.data2 – refers to the data bytes transmitted in D-PHY data lane 2
- tb.csi2_ch0.data3 – refers to the data bytes transmitted in D-PHY data lane 3

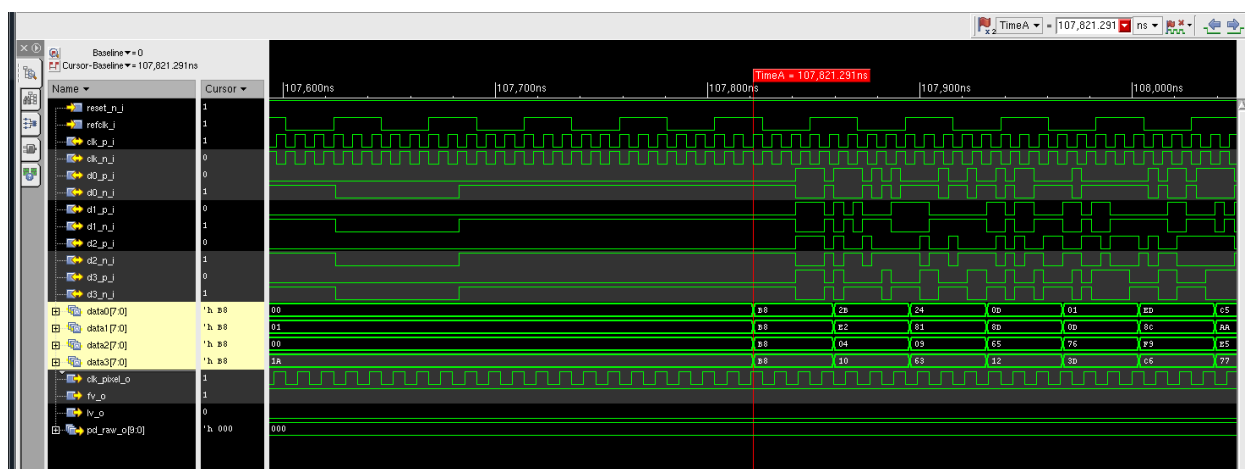


Figure 4.9. CSI-2 Model Video Data

4.8. Instantiating the IP

The core modules of MIPI D-PHY to CMOS Interface Bridge IP are synthesized and provided in NGO format with black box Verilog source files for synthesis. A Verilog source file named `<instance_name>_dphy_2_cmos_ip.v` instantiates the black box of core modules. The top-level file `<instance_name>.v` instantiates `<instance_name>_dphy_2_cmos_ip.v` and PLL component.

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

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.

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 IP is generated. All required Verilog source files for implementation are invoked automatically. The IP can be directly synthesized, mapped and placed/routed in the Diamond design environment after the IP is 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>\dphy2cmos_eval\<instance_name>\impl\lifmd\lse or  
\<project_dir>\dphy2cmos_eval\<instance_name>\impl\lifmd\synplify
```

 directories and set them as active strategy and active preference file.

Push-button implementation of this IP with either Lattice Synthesis Engine (LSE) or Synopsys Synplify Pro RTL synthesis is supported via the pre-built Diamond project file `<instance_name>_top.ldf` located in

```
\<project_dir>\dphy2cmos_eval\<instance_name>\impl\lifmd\lse or  
\<project_dir>\dphy2cmos_eval\<instance_name>\impl\lifmd\synplify
```

 directories.

To use the pre-built Diamond project file:

1. Choose **File > Open > Project**.
2. In the **Open Project** dialog box, browse to
`\<project_dir>\dphy2cmos_eval\<instance_name>\impl\lifmd\<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 MIPI D-PHY to CMOS Interface Bridge IP supports Lattice's IP hardware evaluation capability, which makes it possible to create versions of the IP that operate in hardware for a limited period of time (approximately four hours) 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 instance 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** in the menu as shown in Figure 4.10.

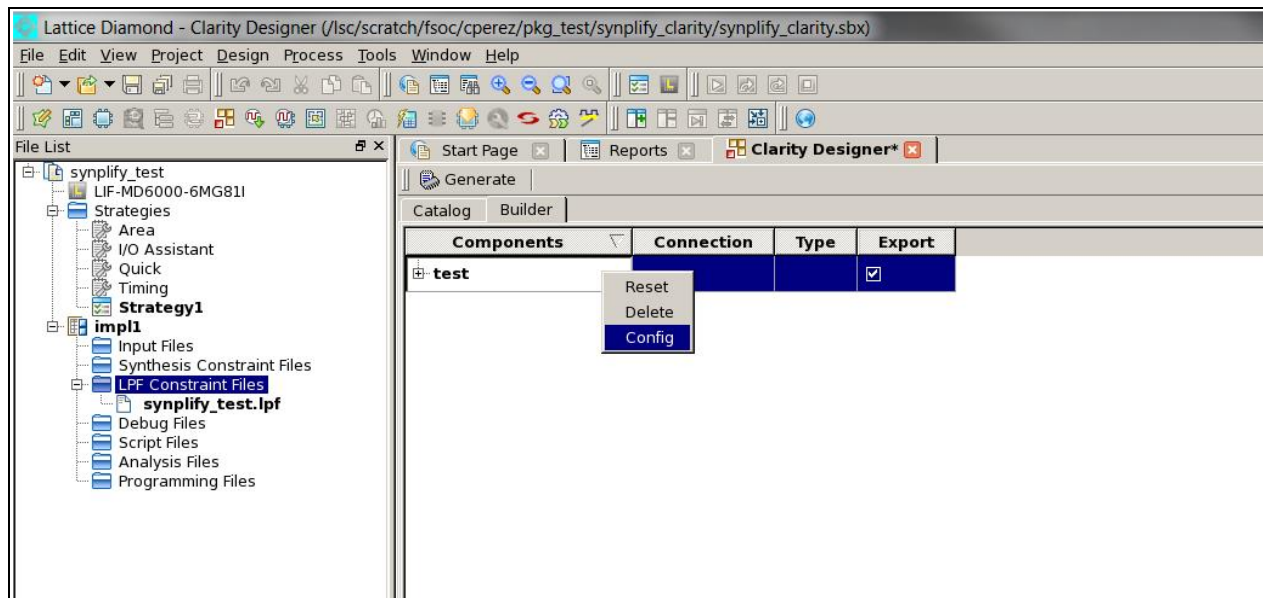



Figure 4.10. Regenerating IP in Clarity Designer

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 IP instances which are reconfigured.

References

For more information about CrossLink devices, refer to the [CrossLink Family Data Sheet \(FPGA-DS-02007\)](#).

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 DSI, 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 for Lattice CrossLink FPGAs using the MIPI D-PHY to CMOS Interface Bridge IP. The performance and utilization data target an LIF-MD6000-6MG81I device with –6 speed grade using Lattice Diamond 3.9 and Lattice Synthesis Engine. Performance may vary when using a different software version or targeting a different device density or speed grade within the CrossLink family. The values of f_{MAX} shown are based on continuous byte clock. The Target f_{MAX} column shows target byte clock frequency for each configuration. See the [Reset and Clocking](#) section for more details on supported clock frequencies.

Table A.1. Resource Utilization

IP User-Configurable Parameters	Slices	LUTs	Registers	sysMEM EBRs	Programmable I/O	Actual f_{MAX} (MHz)	Target f_{MAX} (MHz)
DSI to RGB888, 4-lane, gear16 Non-continuous D-PHY clock	1448	1805	1458	6	30	116.063	93.75
DSI to RGB888, 4-lane, gear8 Non-continuous D-PHY clock	686	819	792	3	30	124.023	112.5
DSI to RGB888, 4-lane, gear16 Continuous D-PHY clock	1441	1794	1451	6	29	116.036	93.75
DSI to RGB888, 4-lane, gear8 Continuous D-PHY clock	679	808	785	3	29	125.329	112.5
CSI-2 to RGB888, 4-lane, gear16 Non-continuous D-PHY clock	1407	1776	1425	6	29	106.849	93.75
CSI-2 to RGB888, 4-lane, gear8 Non-continuous D-PHY clock	669	815	764	3	29	127.486	112.5
CSI-2 to RGB888, 4-lane, gear16 Continuous D-PHY clock	1400	1765	1418	6	28	106.033	93.75
CSI-2 to RGB888, 4-lane, gear8 Continuous D-PHY clock	662	804	757	3	28	129.299	112.5

Appendix B. What is Not Supported

The MIPI D-PHY to CMOS Interface Bridge IP does not support the following features:

- PPI (PHY Protocol Interface)
- Low-level protocol error detection (SoT Error, SoT Sync Error, and so on)
- ECC check and error detection/correction of packet header in a short and a long packet
- Checksum calculation and error detection in long packet
- Command mode operation in MIPI DSI
- Non-burst mode with sync events and burst mode in MIPI DSI
- DCS parsing in MIPI DSI
- CCI communication in MIPI CSI-2
- Optional line packets in MIPI CSI-2

The MIPI D-PHY to CMOS Interface Bridge IP has the following design limitations:

- Maximum value of word count in a long packet is 16'hFFF5
- Minimum duration of MIPI D-PHY low power states (tLPX) should be at least two times the byte clock period when MIPI D-PHY clock lane is continuous and at least two times the reference clock period when MIPI D-PHY clock lane is non-continuous
- Maximum fabric speed is 150 MHz
- Maximum byte clock frequency is 112.5 MHz, lower than maximum fabric speed due to heavy logic inside core modules
- Video VSYNC and HSYNC outputs solely depend on MIPI DSI VSYNC/HSYNC start and end short packets. For displays that require strict timing, design needs to be modified to have additional control

Revision History

Revision 1.4, IP Version 1.2, April 2019

Section	Change Summary
Introduction	Specified that this user guide can be used for IP design version 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 1.3, IP Version 1.2, January 2017

Section	Change Summary
Introduction	<ul style="list-style-type: none"> Updated resource utilization data in Table 1.1. MIPI D-PHY to CMOS Interface Bridge IP Quick Facts in Quick Facts section. Updated maximum data rate supported from 900 Mb/s to 1.5 Gb/s per lane in Features section.
Functional Description	Updated Table 2.3. Clock Frequency Calculations, Table 2.4. Multiplier for Computing Reference Clock Frequency and Table 2.5. Supported Frequencies for MIPI D-PHY to CMOS Interface Bridge IP Configurations to reflect high data rates.
Parameter Settings	Updated Table 3.1. MIPI D-PHY to CMOS Interface Bridge IP Parameter Settings for Rx and Tx gearing.
IP Generation and Evaluation	<ul style="list-style-type: none"> Updated IP name in IP Catalog to <i>d-phy to cmos</i> in Generating IP in Clarity Designer section. Updated Figure 4.1. Clarity Designer Window, Figure 4.3. Configuring MIPI D-PHY to CMOS Interface Bridge IP in Clarity Designer, Figure 4.4. Configuration Tab in IP User Interface, and Figure 4.5. Video Tab in IP User Interface.
Resource Utilization	Updated resource utilization and maximum frequency data with Diamond 3.9 in Table A.1. Resource Utilization.
Appendix B. What is Not Supported	Updated Rx fabric clock limitation from 112 MHz to 112.5 MHz.

Revision 1.2, November 2016

Section	Change Summary
IP Generation and Evaluation	Updated Licensing the IP section – Added email address lic_admn@latticesemi.com for requesting free license.

Revision 1.1, July 2016

Section	Change Summary
All	Updated document number, the previous document number was IPUG123.
Introduction	Updated Synplify Pro version in Table 1.1. MIPI D-PHY to CMOS Interface Bridge IP Quick Facts, and added simulation in Quick Facts section.
Parameter Settings	Updated description of D-PHY Clock Frequency and D-PHY Clock Mode in Table 3.1. MIPI D-PHY to CMOS Interface Bridge IP Parameter Settings.
IP Generation and Evaluation	<ul style="list-style-type: none"> Updated Generated IP Directory Structure and Files section. Added new sections Running Functional Simulation, Simulation Strategies, and Simulation Environment. Updated Instantiating the IP section with instance templates.
Appendix A. Resource Utilization	Updated values of Slices, LUTs, Registers, Actual f_{MAX} in Table A.1. Resource Utilization.

Revision 1.0, IP Version 1.0, May 2016

Section	Change Summary
All	Initial release



www.latticesemi.com