1:2 MIPI DSI Display Interface Bandwidth Reducer IP

User Guide

FPGA-IPUG-02028-1.1

April 2019

Contents

1. Introduction 4
1.1. Quick Facts 4
1.2. Features 4
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. Design and Module Description 9
3. Parameter Settings 10
4. IP Generation and Evaluation 14
4.1. Licensing the IP 14
4.2. Getting Started 14
4.3. Generating IP in Clarity Designer 15
4.4. Generated IP Directory Structure and Files 19
4.5. Running Functional Simulation 20
4.6. Simulation Strategies 22
4.7. Simulation Environment 22
4.8. Instantiating the IP 24
4.9. Synthesizing and Implementing the IP 24
4.10. Hardware Evaluation 24
4.10.1. Enabling Hardware Evaluation in Diamond 24
4.11. Updating/Regenerating the IP 25
4.11.1. Regenerating an IP in Clarity Designer 25
References 26
Technical Support Assistance 26
Appendix A. Resource Utilization 27
Appendix B. What is Not Supported 28
Appendix C. Initializing the DCS ROM 29
Low-Power Mode 29
High-Speed Mode 29
Appendix D. HS Blanking Requirements 32
Revision History 33

Tables

Table 1.1. 1:2 MIPI DSI Display Interface Bandwidth Reducer IP Quick Facts 4
Table 2.1. Top Level Ports 7
Table 2.2. dsi_2_dual_dsi_br_ip Block Modules 9
Table 3.1. 1:2 MIPI DSI Display Interface Bandwidth Reducer IP Parameter Settings 10
Table 4.1. List of Generated Files 19
Table 4.2. Testbench Directives 20
Table 4.3. Testbench Directives for D-PHY Timing Parameters 21
Table 4.4. Design Directives also used by Testbench 22
Table A.1. Resource Utilization 27

Figures

Figure 1.1. Data Ordering 5
Figure 1.2. Left-Right Output Mode 6
Figure 1.3. Odd-Even Output Mode 6
Figure 2.1. MIPI DSI Bandwidth Reducer Block Diagram 7
Figure 4.1. Clarity Designer Window 14
Figure 4.2. Starting Clarity Designer from Diamond Design Environment 15
Figure 4.3. Configuring 1:2 MIPI DSI Display Interface Bandwidth Reducer IP in Clarity Designer 16
Figure 4.4. Configuration Tab in IP Interface 17
Figure 4.5. Initialization Tab in IP Interface 17
Figure 4.6. Protocol Timing Parameters Tab in IP Iinterface 18
Figure 4.7. Miscellaneous Tab in IP Interface 18
Figure 4.8. 1:2 MIPI DSI Display Interface Bandwidth Reducer IP Directory Structure 19
Figure 4.9. Simulation Environment Block Diagram 22
Figure 4.10. PLL Lock and DCS done Miscellaneous Signals 23
Figure 4.11. D-PHY DSI Model Video Data 23
Figure 4.12. Regenerating IP in Clarity Designer 25
Figure C.1. DCS ROM for DCS Low-Power Mode 29
Figure C.2. Sample DCS ROM for x4 Gear 8 DCS High-Speed Mode 30
Figure C.3. Sample DCS ROM for x4 Gear 16 DCS High-Speed Mode 31
Figure C.4. Directory Containing the Sample DCS ROM Initialization Files 31

1. Introduction

The Lattice Semiconductor 1:2 MIPI DSI Display Interface Bandwidth Reducer IP Interfaces a MIPI DSI compliant receiver to two MIPI DSI transmitters through the Lattice Semiconductor CrossLink™ programmable device.

The Mobile Industry Processor Interface (MIPI®) provides specifications for standardization in consumer mobile devices. MIPI Display Serial Interface (DSI) and MIPI D-PHY specifications have been developed to create a standardized interface for all displays used in the mobile industry. As the industry evolves, bandwidth requirements exceed what display manufacturers are capable of manufacturing, while application processor vendors provide very fast interfacing capabilities.

For a cost effective solution, displays can later be replaced with newer display, with the processor retained. Also, multiple displays have gained popularity and extending the output to two display interfaces from a single source becomes a requirement to support these applications. For high-bandwidth application processor interfacing with low resolution displays, the bandwidth can be reduced by distributing the input to multiple displays. The Lattice Semiconductor 1:2 MIPI DSI Display Interface Bandwidth Reducer IP allows you to resolve these interfacing problems.

This user guide is for 1:2 MIPI DSI Display Interface Bandwidth Reducer IP design version 1.x.

1.1. Quick Facts

Table 1.1 provides quick facts about the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP for CrossLink device.

Table 1.1. 1:2 MIPI DSI Display Interface Bandwidth Reducer IP Quick Facts

		1:2 MIPI DSI Display Interface Bandwidth Reducer IP Configuration	
		4-Lane, Left-Right Continuous D-PHY Clock	**4-Lane, Odd-Even, Continuous D-PHY Clock**
IP Requirements	FPGA Families Supported	CrossLink	
Resource Utilization	Targeted Device	LIF-MD6000-6MG81I	
	LUTs	5145	4807
	EBRs	20	4
	Registers	2642	2431
	Programmable I/O	11	11
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 1:2 MIPI DSI Display Interface Bandwidth Reducer IP include:

- Interfaces a MIPI DSI compliant receiver to two MIPI DSI transmitters
- Supports up to 4.8 Gb/s MIPI DSI receive interface
- Supports four data lanes and one clock lane per MIPI DSI interface
- Allows you to store and program a new set of device DCS (Display Command Set)
- Supports RGB888 MIPI DSI video format for left-right output mode
- Supports D-PHY continuous and non-continuous clock modes
- Supports End of Transmission packet (EoTP) generation
- Supports transmission of high-speed blanking of horizontal sync assertion (HSA), horizontal back porch (HBP) and horizontal front porch (HFP) inside active region (VACT)
- Compliant with MIPI D-PHY v1.1 and MIPI DSI v1.1 specifications

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.

Single-bit data stream from each MIPI D-PHY data lane is deserialized into 16-bit parallel data, where bit 0 is the first received bit. The byte in the lower 8 bits of the 16-bit parallel data is the first byte. Figure 1.1 shows byte arrangement within data words.

For left-right output mode, the first half of each video line is transmitted via the first D-PHY transmit channel while the second half of video line is transmitted via the second D-PHY transmit channel. Figure 1.2 shows the left-right output mode.

For odd-even output mode, all odd pixels starting from the first pixel are regrouped and transmitted via the first D-PHY transmit channel, while all even pixels starting from the second pixel are regrouped and transmitted via the second D-PHY transmit channel. Figure 1.3 shows the odd-even output mode.

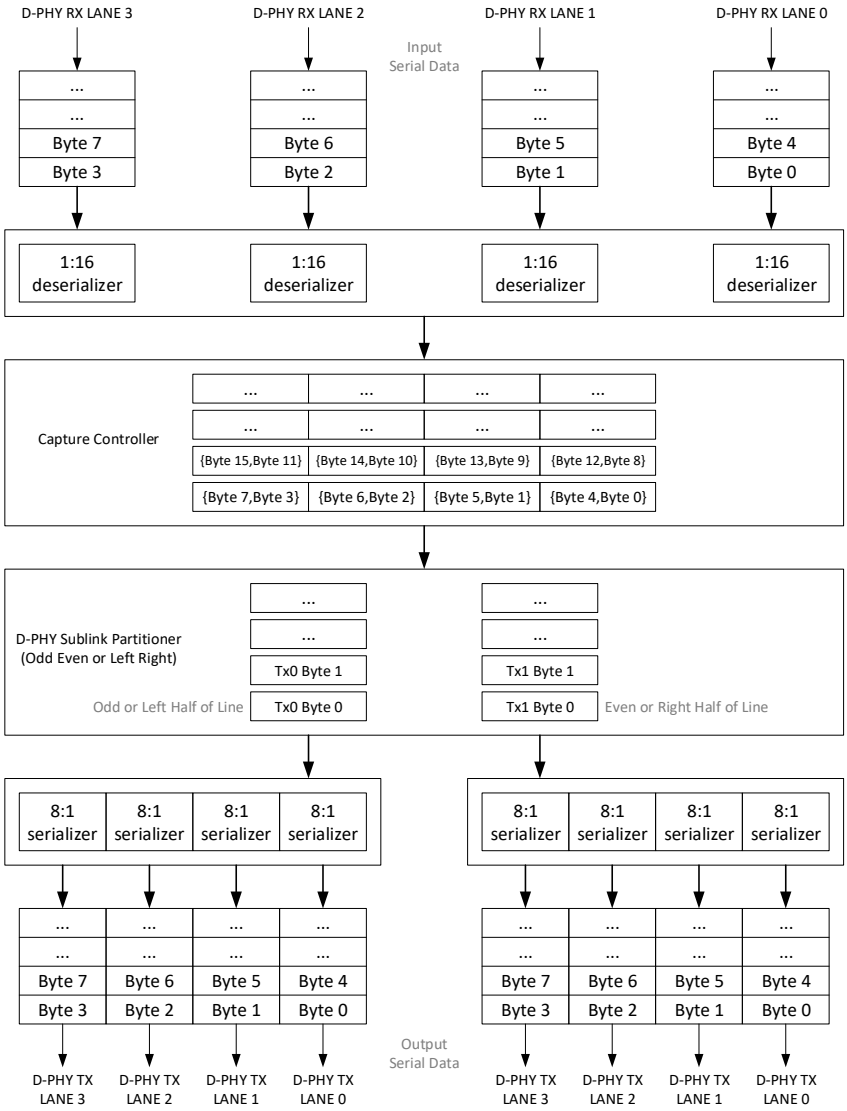

Figure 1.1. Data Ordering

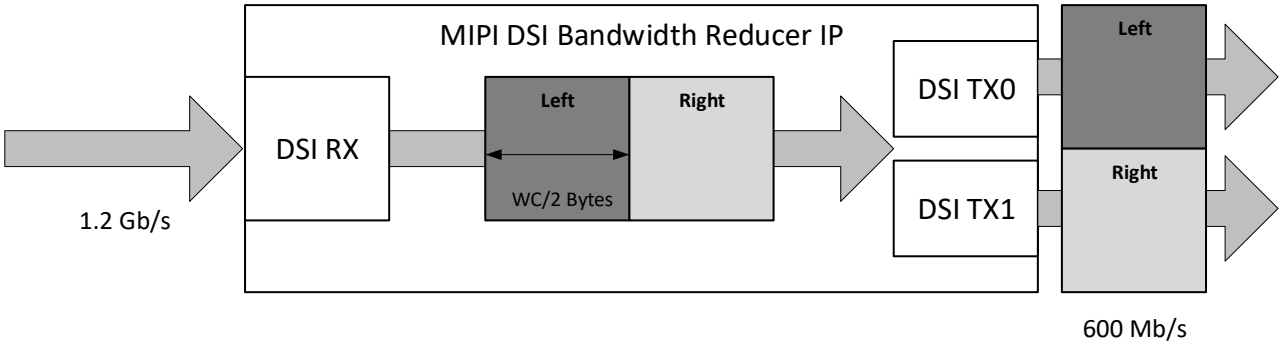

Figure 1.2. Left-Right Output Mode

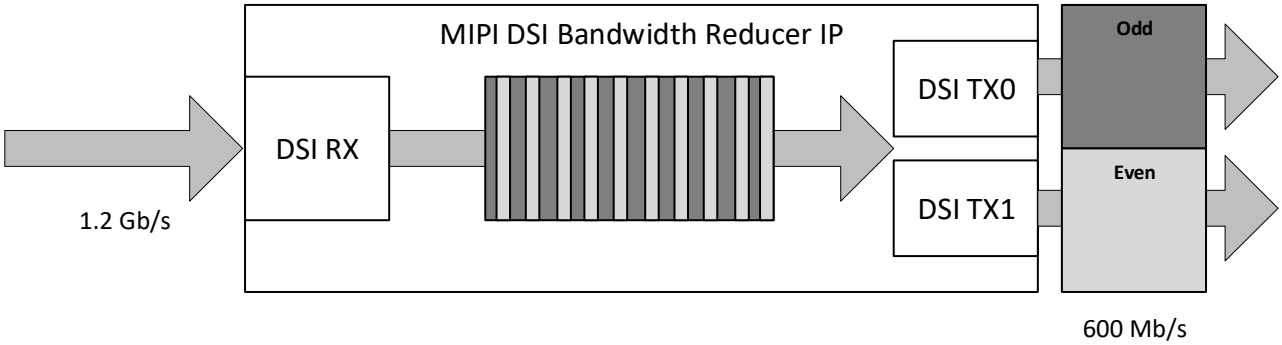

Figure 1.3. Odd-Even Output Mode

1.3.3. Signal Names

Signal names that end with:

- *_i* are input pins.
- *_o* are output pins.
- *_io* are bi-directional pins.
- *_n_i* are active low input signals.

2. Functional Description

Figure 2.1 shows the top level diagram of the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP.

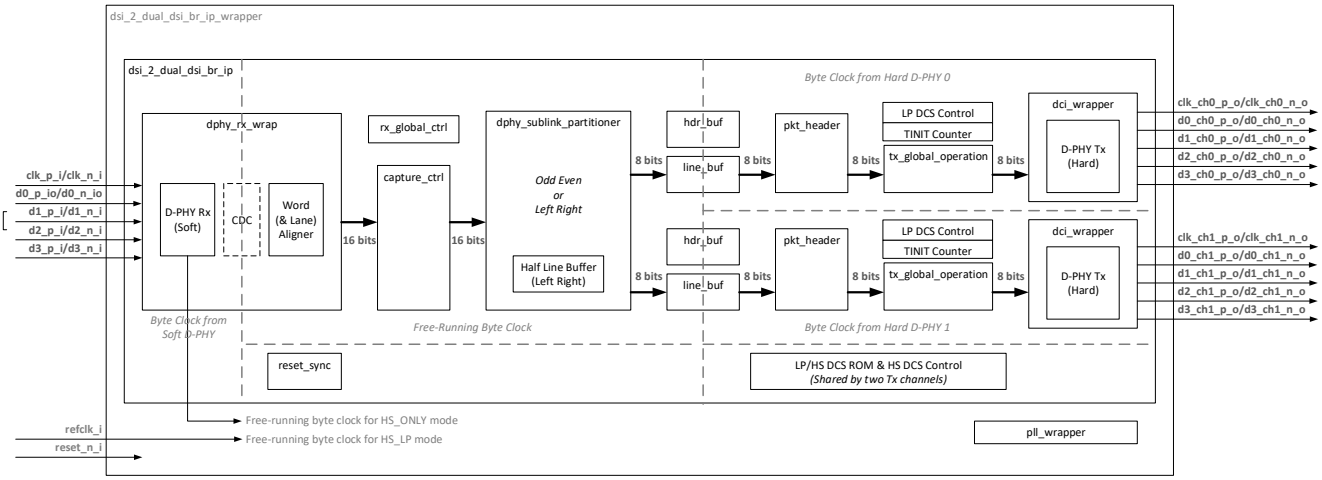

Figure 2.1. MIPI DSI Bandwidth Reducer Block Diagram

Table 2.1 lists the ports of the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP.

Table 2.1. Top Level Ports

Signal	Direction	Functional Description
clk_p_i	I	Positive differential Rx D-PHY input clock
clk_n_i	I	Negative differential Rx D-PHY input clock
d0_p_io	I/O	Positive differential Rx D-PHY input data 0
d0_n_io	I/O	Negative differential Rx D-PHY input data 0
d1_p_i	I	Positive differential Rx D-PHY input data 1
d1_n_i	I	Negative differential Rx D-PHY input data 1
d2_p_i	I	Positive differential Rx D-PHY input data 2
d2_n_i	I	Negative differential Rx D-PHY input data 2
d3_p_i	I	Positive differential Rx D-PHY input data 3
d3_n_i	I	Negative differential Rx D-PHY input data 3
clk_ch0_p_o	O	Positive differential Tx D-PHY output clock, channel 0
clk_ch0_n_o	O	Negative differential Tx D-PHY output clock, channel 0
d0_ch0_p_io	I/O	Positive differential Tx D-PHY output data 0, channel 0
d0_ch0_n_io	I/O	Negative differential Tx D-PHY output data 0, channel 0
d1_ch0_p_o	O	Positive differential Tx D-PHY output data 1, channel 0
d1_ch0_n_o	O	Negative differential Tx D-PHY output data 1, channel 0
d2_ch0_p_o	O	Positive differential Tx D-PHY output data 2, channel 0
d2_ch0_n_o	O	Negative differential Tx D-PHY output data 2, channel 0
d3_ch0_p_o	O	Positive differential Tx D-PHY output data 3, channel 0
d3_ch0_n_o	O	Negative differential Tx D-PHY output data 3, channel 0
clk_ch1_p_o	O	Positive differential Tx D-PHY output clock, channel 1
clk_ch1_n_o	O	Negative differential Tx D-PHY output clock, channel 1
d0_ch1_p_io	I/O	Positive differential Tx D-PHY output data 0, channel 1
d0_ch1_n_io	I/O	Negative differential Tx D-PHY output data 0, channel 1
d1_ch1_p_o	O	Positive differential Tx D-PHY output data 1, channel 1
d1_ch1_n_o	O	Negative differential Tx D-PHY output data 1, channel 1

Signal	Direction	Functional Description
d2_ch1_p_o	O	Positive differential Tx D-PHY output data 2, channel 1
d2_ch1_n_o	O	Negative differential Tx D-PHY output data 2, channel 1
d3_ch1_p_o	O	Positive differential Tx D-PHY output data 3, channel 1
d3_ch1_n_o	O	Negative differential Tx D-PHY output data 3, channel 1
refclk_i	I	Input reference clock for Non-continuous Rx clock mode
reset_n_i	I	Asynchronous active low system reset
d0_ch1_n_io	I/O	Negative differential Tx D-PHY output data 0, channel 1
d1_ch1_p_o	O	Positive differential Tx D-PHY output data 1, channel 1
Miscellaneous Debug Ports		
tx0_dcsrom_done	O	Indicates the DCS initialization of Tx channel 0 is done
tx0_tinit_done	O	Indicates the Initialization delay counter from Tx channel 0 is done
tx0_pll_lock	O	PLL lock indicator for Tx channel 0
tx0_byteclock	O	Tx channel 0 output byte clock from D-PHY PLL
tx0_lp_clk_en	O	Low-power clock enable of Tx channel 0
fifo0_empty	O	Indicates that synchronizing FIFO (line buffer) for Tx channel 0 is empty
tx1_dcsrom_done	O	Indicates the DCS initialization of Tx channel 1 is done
tx1_tinit_done	O	Indicates the Initialization delay counter from Tx channel 1 is done
tx1_pll_lock	O	PLL lock indicator for Tx channel 1
tx1_byteclock	O	Tx channel 1 output byte clock from D-PHY PLL
tx1_lp_clk_en	O	Low-power clock enable of Tx channel 1
fifo1_empty	O	Indicates that synchronizing FIFO (line buffer) for Tx channel 1 is empty

2.1. Design and Module Description

The top module instantiates dsi_2_dual_dsi_br_ip module that contains all major blocks used. If required, the top wrapper also instantiates GPLL to generate x2/x4/x5 reference clock for D-PHY PLL depending on byte clock frequency.

Table 2.2 lists the modules within the dsi_2_dual_dsi_br_ip block.

Table 2.2. dsi_2_dual_dsi_br_ip Block Modules

Module	Description
dphy_2_cmos_ip	• Instantiates the MIPI D-PHY Rx soft IP wrapper. The D-PHY wrapper uses CrossLink DDR I/O and fabric to receive MIPI serial data. This converts the incoming serial data from the D-PHY data lanes to 64-bit (4 lane x16) words. • Instantiates the Rx global controller that contains finite state machines (FSMs) to detect the state transitions of the MIPI D-PHY clock and data lanes. • Instantiates Capture Controller that decodes MIPI DSI packets.
dphy_sublink_partitioner	• Partitions receive video line into left and right half of line or odd and even pixels, and generate 32-bit (4-lane x8) words per Tx channel. • For left-right mode, a line buffer is instantiated to store first half of line and transmit left and right at the same time. The buffer depth must be enough to store receive payload. Maximum buffer size is 2048x64, that is maximum receive word count of ‘d16384.
hdr_line_buf	• Uses two instances of 4x21 FIFO for header buffering and synchronization to Tx byte clocks. • Uses two instances of 512x32 FIFO for line buffering and synchronization to Tx byte clocks. • The MIPI D-PHY specification does not allow for flow control, so the Rx and the Tx byte clock must be equal.
cmos_2_dphy_ip	• There are two instances of this IP block, one for each Tx channel. Each instance contains the hard MIPI D-PHY Tx wrapper that serializes the packet data. Each wrapper contains its own Tx PLL. • This module contains the Tx Global Control module that controls the transitions of the Tx clock and data lanes. • Also included in this module is the low-power DCS Controller. This is disabled if high-speed DCS is used. • It instantiates tinit_counter that drives LP-11 to each D-PHY Tx lane for a configurable number of byte clock cycles starting from D-PHY PLL lock detection.
reset_sync	This module synchronizes system reset signal to different clock domains used in the design.
dcs_rom or dcs_rom_hs	The DCS ROM contains the Display Command Set for the DSI slave.
DCS_hs	This DCS controller is used for high-speed DCS mode.

3. Parameter Settings

Table 3.1 lists the parameters that can be set to configure the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP when the IP is not packaged.

Table 3.1. 1:2 MIPI DSI Display Interface Bandwidth Reducer IP Parameter Settings

Parameter	Attribute	Options	Description
Number of Rx Channels	Fixed	1	This selects the number of MIPI D-PHY receivers.
Number of Rx Lanes	Fixed	4	This selects the number of MIPI D-PHY Rx data lanes.
Rx Gear	Fixed	16	Input serial bits are converted to 16-bit data bus.
Rx D-PHY IP	Fixed	Soft D-PHY	This selects the soft D-PHY Rx logic. The maximum line rate when Soft IP is selected is 1.2 Gb/s. The hard D-PHY option is not available, both available hard D-PHY channels are used in transmit mode.
Lane Aligner FIFO Type	Disabled	EBR or LUT	Disabled, because lane aligner option is not available.
Enable Lane Aligner	Disabled	ON or OFF	Lane aligner is disabled, because available resource is not sufficient. Word aligner is sufficient to handle data lane skews within 1 byte clock period.
Number of Tx Channels	Fixed	2	This selects the number of MIPI D-PHY transmitters. The received line is split into left and right or odd and even and transmitted into two Tx channels. The first Tx channel (TX0) transmits either left part of line or odd pixels. The second Tx channel transmits either right part of line or even pixels. The two channels are asynchronous with each other.
Number of Tx Lanes	Fixed	4	This selects the number of MIPI D-PHY Tx data lanes.
Tx Gear	Fixed	8	Hard D-PHY converts byte data are converted to 1-bit serial data.
Rx Line Rate	User-configurable	320 - 1200	Data rate per Rx lane in Mb/s. Note that T_{LPX}, that is the D-PHY low-power state period, must be twice the byte clock period.
Rx D-PHY Clock Frequency	Read Only	(Rx Line Rate) / 2	The Rx D-PHY Clock Frequency is half of the Rx line rate, in MHz.
Rx D-PHY Clock Mode	User-configurable	Continuous or Non-Continuous	In continuous mode, the input D-PHY clock lane is always in high-speed mode. The 1:2 MIPI DSI Display Interface Bandwidth Reducer IP utilizes this clock to generate the byte clock for internal logic. In non-continuous mode, the input D-PHY clock lane goes to low-power states. So, an external clock, refclk_i, is needed.
Tx D-PHY Clock Mode	Read Only	Continuous or Non-Continuous	Tx D-PHY Clock Mode is the same as Rx D-PHY Clock Mode.
Byte Clock Frequency	Read Only	(Rx Line rate) / (Rx Gear)	This is the frequency that the internal logic operates at.
Reference Clock Frequency	Read Only	(Rx Line rate) / (Rx Gear)	This clock is used to clock the Rx byte clock domain if the Rx D-PHY Clock Mode is non-continuous.
Virtual Channel ID (Ch 0)	User-configurable	00, 01, 10, 11	Virtual channel for Tx channel 0 in two-bit binary format.

Parameter	Attribute	Options	Description
Virtual Channel ID (Ch 1)	User-configurable	00, 01, 10, 11	Virtual channel for Tx channel 1 in two-bit binary format.
Enable EoTP Transmission	User-configurable	ON or OFF	This enables the end of transmit packet (EoTP) generation.
Operation Options	User-configurable	Left-Right or Odd-Even	This selects the output mode. Left-Right splits the received line in half, transmitting the first half through Tx channel 0 and the second half through Tx channel 1 simultaneously. Odd-Even splits the received line into odd and even pixels, transmitting the odd pixels through Tx channel 0 and the even pixels through Tx channel 1 simultaneously.
Rx Word Count	User-configurable	(Integer)	Rx word count is the number of bytes in the Rx payload. The Tx word count is half the Rx word count. As Rx serial data from the four MIPI D-PHY lanes is converted into 64-bit data bus, the Rx word count must be a multiple of 8 bytes so it can evenly be divided into two 32-bit data bus, each going to one of the Tx channels. Rx word count parameter is used in Left-Right mode to determine RAM size. The maximum Rx word count is limited to 11384 due to the number of EBRs available in the device. It is also used if HS blanking is selected to monitor FIFO read count inside line buffer.
Data Type	Fixed	RGB888	The data type parameter is used in Odd=Even mode. Only RGB888 is supported.
RGB666 Type	Disabled	Packed or Loosely Packed	This option is disabled, because only RGB888 is supported.
Horizontal Sync Active (HSA)*	User-configurable	(Integer)	Blanking packet word count in between HSYNC start and HSYNC end short packets in transmit side. The value must be 2 + multiple of 4. The minimum HSA supported is 26.
Horizontal Back Porch (HBP)*	User-configurable	(Integer)	Blanking packet word count in between HSYNC end short packet and active data long packet in transmit side. The value must be 2 + multiple of 4. The minimum HBP supported is 38.
Horizontal Front Porch (HFP)*	User-configurable	(Integer)	Blanking packet word count in between active data long packet and HSYNC start short packet in transmit side. The value must be a multiple of 4. The minimum HFP supported is 28.
Lines during VSYNC region (VSA)	User-configurable	(Integer)	This specifies number of lines during VSYNC Active (VSA) region.
Vertical Back Porch (VBP)	User-configurable	(Integer)	This specifies number of lines during Vertical Back Porch (VBP) region.
Active lines per frame (VACT)	User-configurable	(Integer)	This specifies number of lines during Vertical Active (VACT) region.
Enable high-speed blanking during VACT	User-configurable	ON or OFF	This enables HS blanking feature. HS blanking is provided to support shorter blanking periods that cannot be supported with LP blanking option. Note that only Non-burst mode with Sync Pulses DSI video mode is supported when HS blanking is selected.
Number of DCS Words	User-configurable	10-bit non-zero decimal value	This defines the number of valid words in the DCS ROM initialization file, including the sync pattern and the trail bytes.

Parameter	Attribute	Options	Description
DCS ROM Wait Time	User-configurable	1–4096	This parameter sets the interval between DCS packets in terms of number of byte clock cycles. This applies to both high-speed and low-power DCS timing mode.
DCS Mode	User-configurable	Low power or High Speed	DCS initialization of the DSI slave may be performed in D-PHY low-power timing mode, or in high-speed mode. If HS DCS is selected, only one DCS ROM is used for both Tx channels to save LUT resources. DCS for Tx channel 0 is transmitted first followed by DCS for Tx channel 1.
DCS ROM Initialization File	User-configurable	Text file	This must be a text file that contains the display command set data packets. See Appendix C. *Initializing the DCS ROM* for the format of entries.
Bypass DCS	User-configurable	ON or OFF	You can bypass DCS feature and send DCS from AP through bridge under normal operation.
tINIT_SLAVE Value	User-configurable	16-bit non-zero decimal value	This parameter, in addition to the DCS ROM Wait Time parameter, sets the period needed to meet the required initialization time of the DSI slave. The value is in terms of byte clock cycles. The D-PHY specification places a minimum period of 100 µs, but this parameter may be increased depending on the receiver requirement. During this period, all incoming data is ignored by the bridge.
Bypass tINIT counter	User-configurable	ON or OFF	You can bypass tINIT counter as PLL lock time takes around 15 ms, more than enough to meet D-PHY tINIT requirement.
t_HS-PREPARE	User-configurable	1–99	This sets the $T_{HS-PREPARE}$ counter in terms of number of byte clock cycles. It is the time the transmitter drives the Data Lane LP-00 Line state immediately before the HS-0 Line state starting the HS transmission.
t_HS-ZERO	User-configurable	1–99	This sets the $T_{HS-ZERO}$ counter in terms of number of byte clock cycles. It is the time the transmitter drives the HS-0 state prior to transmitting the Sync sequence. In gear 8, the actual $T_{HS-ZERO}$ has ~2.5 cycles more than the specified value due to the register delays when converting data from parallel to serial. In gear 16, the actual value has ~3.5 cycles more.
t_CLK-PRE	User-configurable	1–99	This sets the $T_{CLK-PRE}$ counter in terms of number of byte clock cycles. This is the time that the transmitter drives the HS clock prior to any associated Data Lane beginning the transition from LP to HS mode. The actual $T_{CLK-PRE}$ has 2 more additional byte clock cycles due to register delays.
t_CLK-POST	User-configurable	1–99	This sets the $T_{CLK-POST}$ counter in terms of number of byte clock cycles. This is the time that the transmitter continues to send HS clock after the last associated Data Lane has transitioned to LP Mode. Interval is defined as the period from the end of $T_{HS-TRAIL}$ to the beginning of $T_{CLK-TRAIL}$. The actual $T_{CLK-POST}$ has one more additional byte clock cycle due to register delays.

Parameter	Attribute	Options	Description
Enable Miscellaneous Status Signals	User-configurable	ON or OFF	Enabling the miscellaneous signals ports out some internal signals for debug purposes.
Enable Tx0 DCS done Enable Tx1 DCS done Enable Tx0 tINIT done Enable Tx1 tINIT done Enable Tx0 PLL lock Enable Tx1 PLL lock Enable Tx0 Byte clock Enable Tx1 Byte clock Enable Tx0 LP Clock Enable Enable Tx1 LP Clock Enable Enable Tx0 FIFO Empty Enable Tx1 FIFO Empty	User-configurable	ON or OFF	Each miscellaneous signal can be enabled/disabled separately.

***Note**: HSA, HBP and HFP parameters for HS blanking feature are word counts for blanking packet generation. To properly match Rx and Tx line times and avoid FIFO overflow/underflow inside line buffers, the requirements described in Appendix D must be followed.

4. IP Generation and Evaluation

This section provides information on how to generate 1:2 MIPI DSI Display Interface Bandwidth Reducer IP using the Lattice Diamond Clarity Designer, and how to run simulation, synthesis and hardware evaluation.

4.1. Licensing the IP

The 1:2 MIPI DSI Display Interface Bandwidth Reducer IP license is available free of charge, but an IP-specific license is required to enable full, unrestricted use of the 1:2 MIPI DSI Display Interface Bandwidth Reducer 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 IP for your design.

You may download and generate the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP and fully evaluate through functional simulation and implementation (synthesis, map, place and route) without an IP license. The 1:2 MIPI DSI Display Interface Bandwidth Reducer IP also supports Lattice's 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 1:2 MIPI DSI Display Interface Bandwidth Reducer 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 Design interface as shown in Figure 4.1.

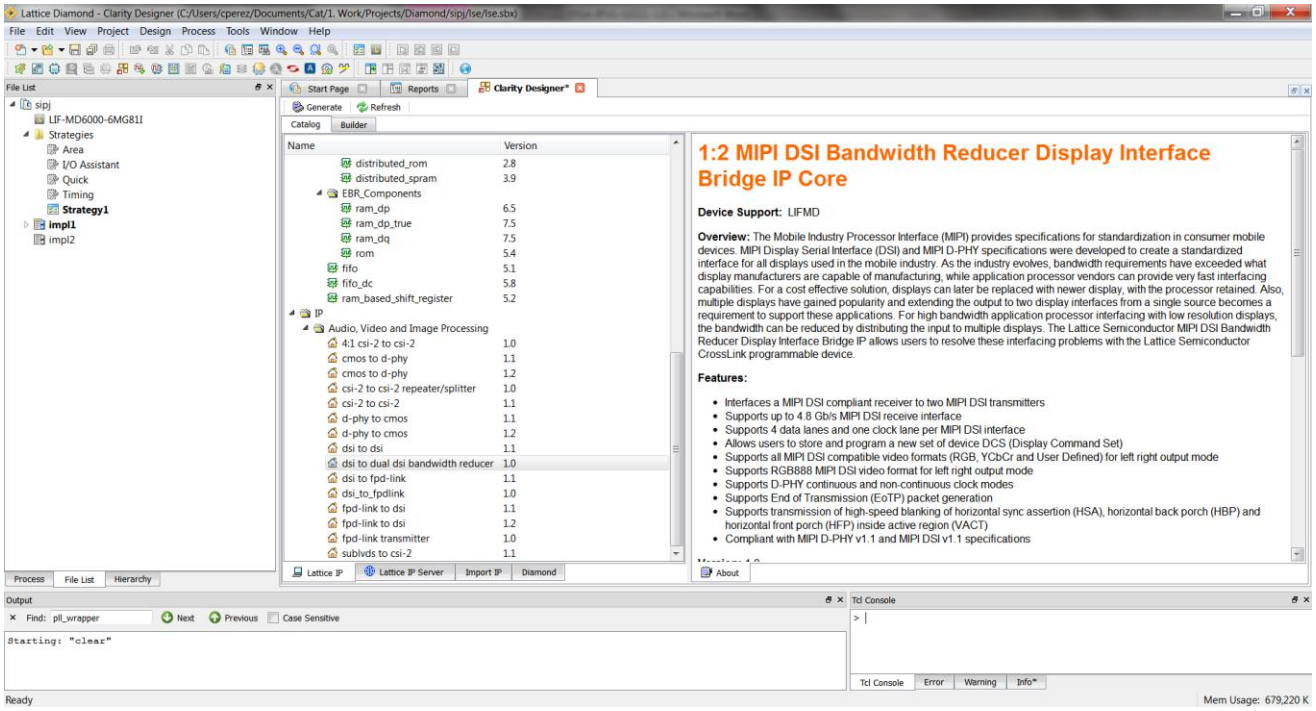

Figure 4.1. Clarity Designer Window

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. 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 1:2 MIPI DSI Display Interface Bandwidth Reducer 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 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** – Click this to create a new Clarity Design project directory in which the 1:2 MIPI DSI Display Interface Bandwidth Reducer 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.

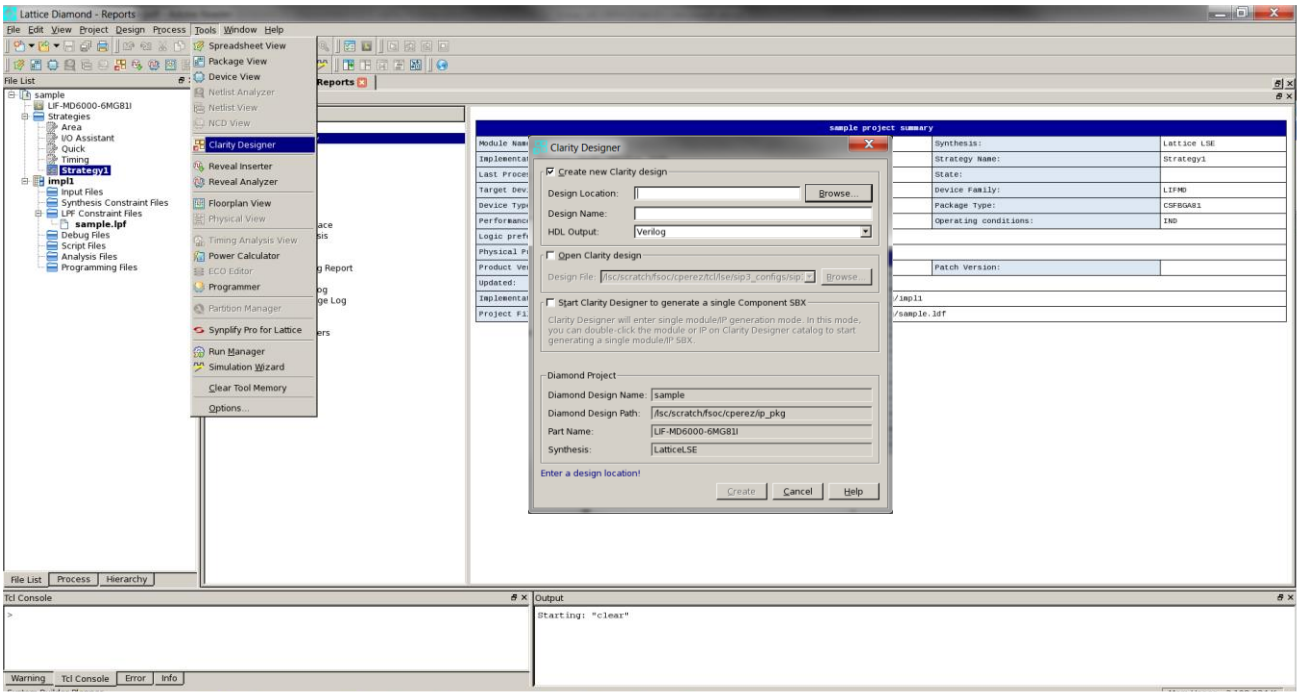

Figure 4.2. Starting Clarity Designer from Diamond Design Environment

To configure the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP in Clarity Designer:

1. Double-click **dsi to dual dsi bandwidth reducer** in the IP list of the Catalog view. The **dsi to dual dsi bandwidth reducer** dialog box is displayed as shown in Figure 4.3.

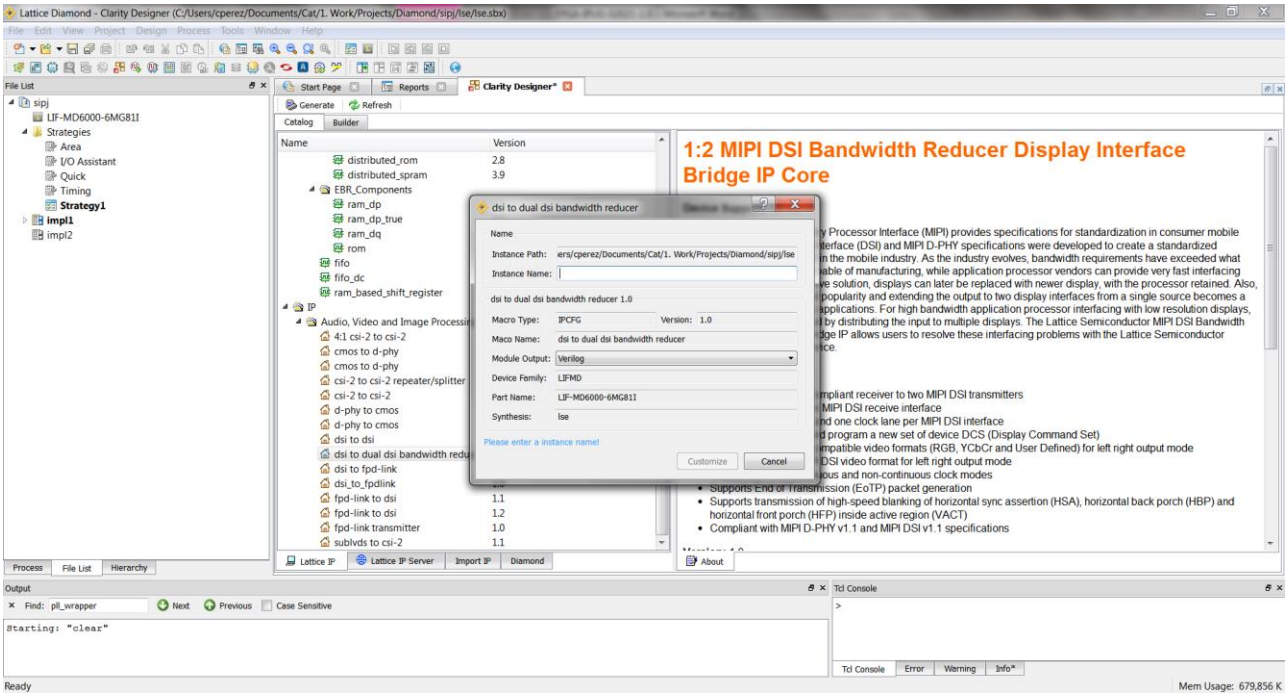

Figure 4.3. Configuring 1:2 MIPI DSI Display Interface Bandwidth Reducer IP in Clarity Designer

2. Enter the **Instance Name**.
3. Click the **Customize** button. An IP configuration user interface is displayed as shown in Figure 4.4 to Figure 4.7. From this dialog box, you can select the IP parameter options specific to your application.
4. Select the required parameters, and click the **Configure** button.
5. Click **Close**.
6. 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.

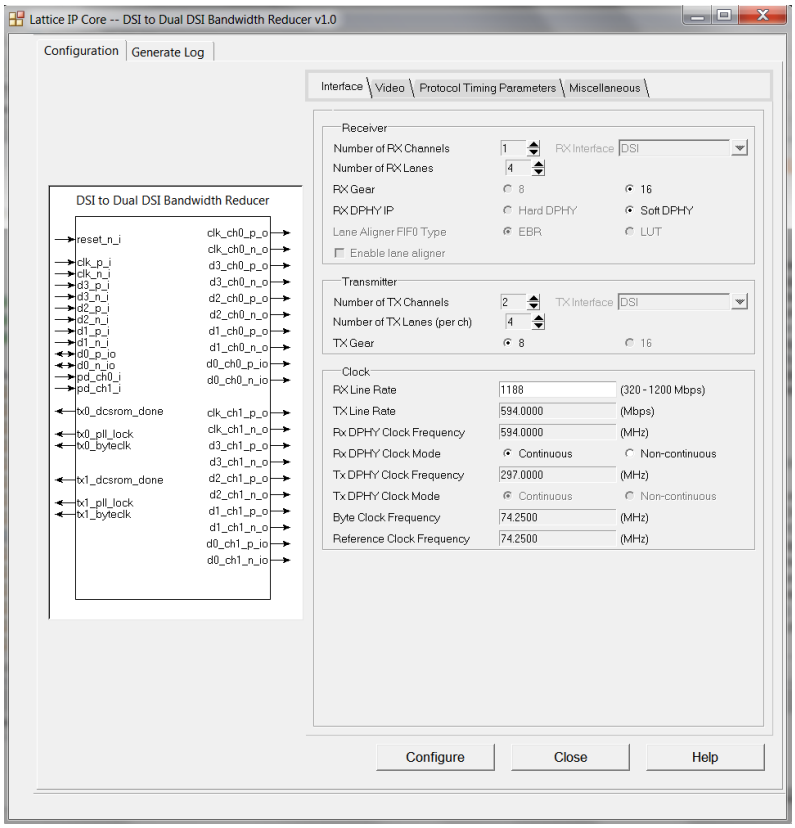

Figure 4.4. Configuration Tab in IP Interface

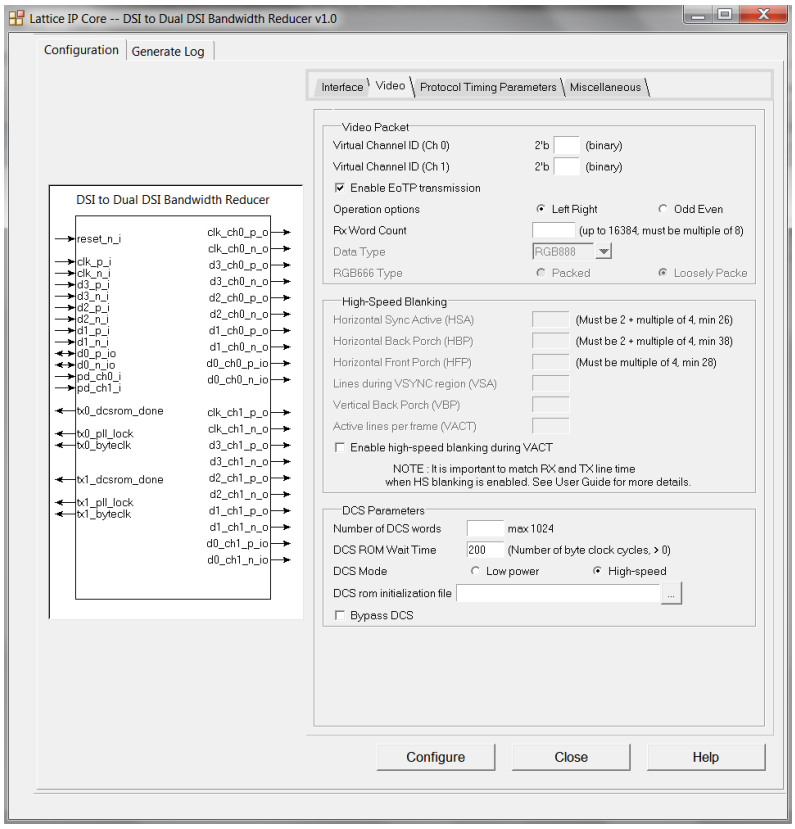

Figure 4.5. Initialization Tab in IP Interface

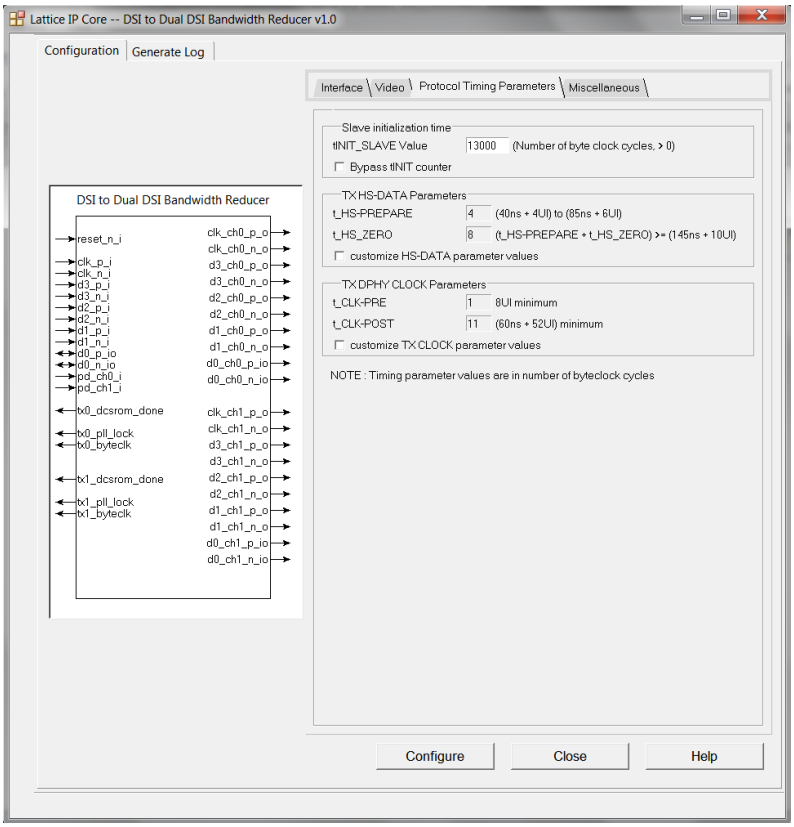

Figure 4.6. Protocol Timing Parameters Tab in IP Iinterface

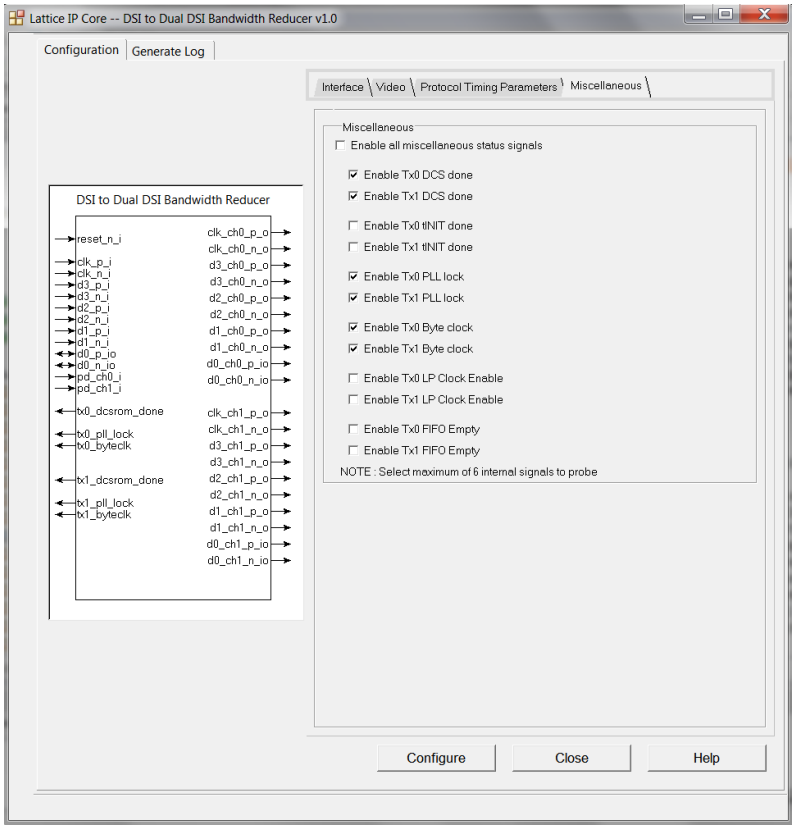

Figure 4.7. Miscellaneous Tab in IP Interface

4.4. Generated IP Directory Structure and Files

Figure 4.8 shows the directory structure of generated IP files.

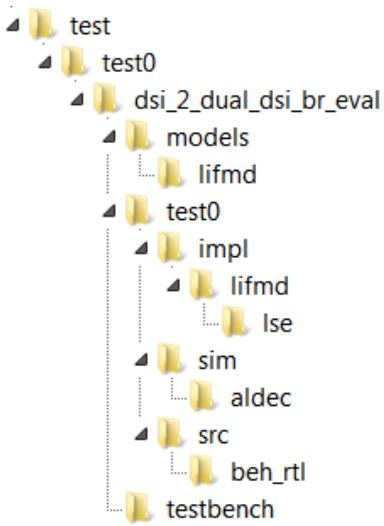

Figure 4.8. 1:2 MIPI DSI Display Interface Bandwidth Reducer IP Directory Structure

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

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

Table 4.1. List of Generated Files

File	Description
<instance_name>.v	Verilog top-level module of MIPI DSI to Dual DSI Bandwidth Reducer 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>_inst.v/vhd	Template for instantiating the design in another user-created top module
<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 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.

All IP files are generated inside \<*project_dir*> directory (test folder in Figure 4.8). The \<*project_dir*> is composed of <*design_location*>\<*design_name*>\<*instance_name*>, see the Generating IP in Clarity Designer section. A separate \<*project_dir*> is created each time 1:2 MIPI DSI Display Interface Bandwidth Reducer IP is created with a different IP instance name.

The *dsi_2_dual_dsi_br_eval* and subdirectories provide files supporting push-button IP evaluation through functional simulations, design implementation (synthesis, map, place and route) and hardware evaluation. Inside *dsi_2_dual_dsi_br_eval is* \<*instance_name*> folder (test0 folder in Figure 4.8) 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 that you specified in Clarity Designer. The simulation part of user evaluation provides testbench and test cases supporting RTL simulation for Active-HDL simulator under \<*project_dir*>*testbench*. Separate directories located at \<*project_dir*>*dsi_2_dual_dsi_br_eval*\<*instance_name*>*sim**aldec* are provided and contain specific pre-built simulation script files. See the Running Functional Simulation section below for details.

4.5. Running Functional Simulation

To run simulations using Active-HDL:

1. Create new project using Lattice Diamond for Windows.
2. Open **Active-HDL** Lattice Edition interface tool.
3. Modify the ***.do** file located in **\<project_dir>\<core_instance_name>\<core_name>_eval\<core_instance_name>\sim\aldec**.
   a. Specify the working directory (sim_working_folder). For example, set sim_working_folder **C:/my_design**.
   b. Specify the workspace name that is created in working directory. For example, set workspace_name **design_space**.
   c. Specify the design name. For example, set design_name **DesignA**.
   d. Specify the design path where the IP Core generated using Clarity Designer is located. For example, set design_path **C:/my_designs/DesignA**.
   e. Specify the design instance name (same as the instance name specified in Clarity Designer). For example, set design_inst **DesignA_inst**.
   f. Specify the Lattice Diamond Primitive path (diamond_dir) to where it is installed. For example, set diamond_dir **C:/lscc/diamond/3.8_x64**.
   g. Update the testbench parameters to customize data size, clock, and/or other settings. See Table 4.2 and Table 4.3 for the list of valid testbench compiler directives.
4. Click **Tools** -> **Execute Macro**, then select the ***.do** file.
5. Wait for the simulation to finish.

Table 4.2 lists the testbench directives which can be modified by setting the define in the vlog command in ***.do** file.

Example:

```
vlog \
+define+NUM_FRAMES=60 \
…
```

Table 4.2. Testbench Directives

Directive	Description
PLL_DURATION , tINIT_DURATION, DCS_DURATION	Used when miscellaneous signals are off (for example, debug output ports for PLL lock, tINIT done, and/or DCS ROM done are not included in the generated design). This directive is used to set the duration of PLL lock (in ps), tINIT done, and/or DCS ROM done before the D-PHY model in the testbench transmits input data to the design. Example: +define+PLL_DURATION=15000000 +define+tINIT_DURATION=100000000 +define+DCS_DURATION=950000000
NUM_FRAMES	Sets the number of video frames
NUM_LINES	Sets the number of lines per frame
FRAME_LPM_DELAY	Sets the low-power mode delay between frames (in ps)
VIDEO_DATA_TYPE	Video data type, in decimal value. For example, for RGB888 (0x3E), +define+VIDEO_DATA_TYPE=62
VFP_LINES	Number of Vertical Front Porch Lines

Directive	Description
EOTP_ENABLE	Used to enable/disable transmission of End-of-Transmit packet 0 – EoTP packet is disabled 1 – EoTP packet is enabled
BLLP_PAYLOAD	Number of bytes of BLLP Payload (used for HS data blanking)
LPS_BLLP_DURATION	Used to set the duration (in ps) for BLLP low-power state (used for LP blanking)
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)
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)
VIRTUAL_CHANNEL	Used to set the virtual channel number
NON_BURST_SYNC_EVENTS BURST_MODE NON_BURST_SYNC_PULSE	Video Mode Types. One of the following video mode types must be defined. The default mode used by the testbench is Non-burst sync pulse. For example add +define+BURST_MODE in vlog command to enable Burst Mode
DPHY_DEBUG_ON	Used to enable or disable debug messages 0 – Debug messages are disabled 1 – Debug messages are enabled
DPHY_CLK_PERIOD	By default, the testbench automatically calculates the D-PHY clock period, but you can change/override the clock period by defining the directive in vlog (in ps). For example +define+DPHY_CLK_PERIOD=1684
REFCLK_PERIOD	By default, the testbench automatically calculates the reference clock period for Non-Continuous Rx Clock Mode, but you can change/override the clock period by defining the directive in vlog (in ps). For example +define+REFCLK_PERIOD=6736
HS_BLANKING	By default, low-power blanking is used during HS_LP mode. To use HS data blanking, HS_BLANKING may be added in the list of defines (+define+HS_BLANKING)
LP_BLANKING	By default, HS data blanking is used during HS_ONLY mode. To use low-power blanking, LP_BLANKING may be added in the list of defines (+define+LP_BLANKING)

The testbench has default setting for the D-PHY timing parameters listed in Table 4.3. If required you can modify the D-PHY timing parameters by setting the following 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)

Refer to *MIPI D-PHY Specification version 1.1, Table 14* for information regarding D-PHY timing requirements.

The testbench also uses the design parameters listed in Table 4.4.

Table 4.4. Design Directives also used by Testbench

Directive	Description
VACT	Number of bytes of active pixels per line
HSA	Number of bytes of Horizontal Sync Active Payload (used for Non-burst sync pulse)
HBP	Number of bytes of Horizontal Back Porch Payload (used for HS data blanking, and in LP blanking for Non-burst sync pulse mode)
HFP	Number of bytes of Horizontal Front Porch Payload (used for HS data blanking, and in LP blanking for Non-burst sync pulse mode)
VSA	Number of Vertical Sync Active Lines
VBP	Number of Vertical Back Porch Lines

4.6. Simulation Strategies

This section describes the simulation environment which demonstrates basic 1:2 MIPI DSI Display Interface Bandwidth Reducer IP functionality. Figure 4.9 shows the block diagram of simulation environment.

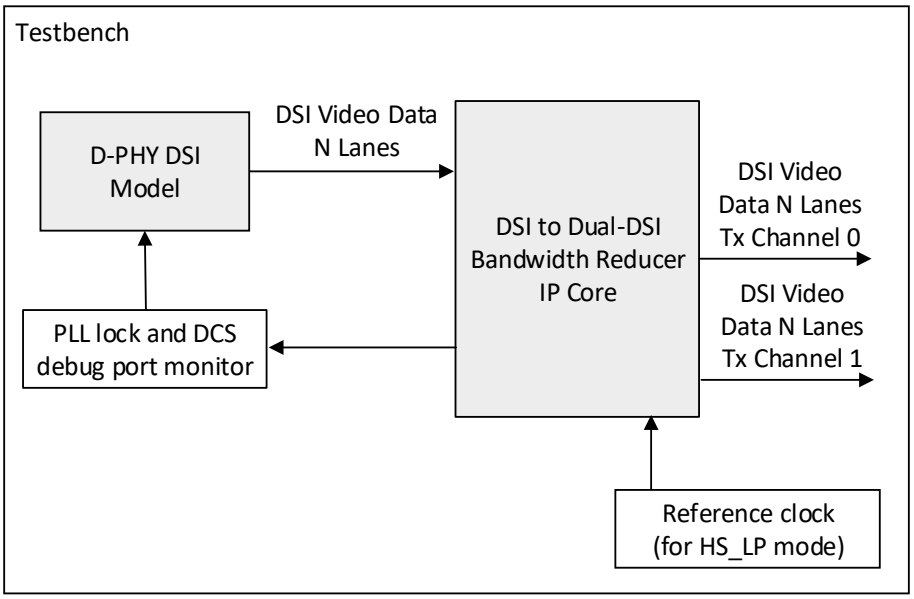

Figure 4.9. Simulation Environment Block Diagram

4.7. Simulation Environment

The simulation environment is made up of the D-PHY DSI model instance connected to the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP core instance in the testbench. The D-PHY DSI model is configured based from the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP core configurations and testbench configurations. The testbench also transmits reference clock to the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP core if clock mode is non-continuous (HS_LP). If miscellaneous signals, such as PLL lock and DCS done debug ports, are included in the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP core, the testbench monitors assertion of these signals before sending the DSI video data to the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP core.

Figure 4.10. shows an example simulation where PLL lock and DCS done debug ports are included. Note that for two Tx channels, the Tx channel 0 HS DCS is done first before Tx channel 1 HS DCS. Tx channel 1 may output the data at an earlier time if it detects the input data immediately after HS DCS is done as shown in the figure at Time A.

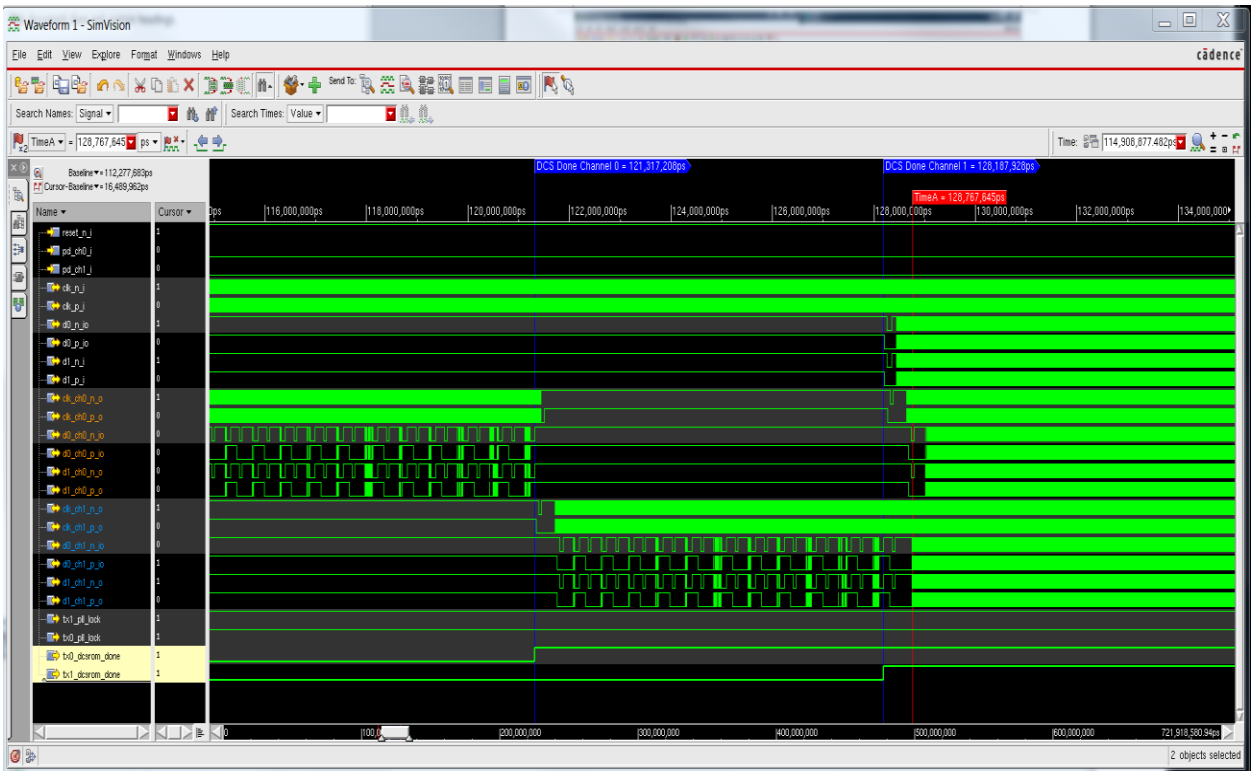

Figure 4.10. PLL Lock and DCS done Miscellaneous Signals

The video data transmitted by the D-PHY DSI model can viewed in waveform, see Figure 4.11:

- tb.dphy_ch0.data0 – refers to the data bytes transmitted in D-PHY data lane 0
- tb.dphy_ch0.data1 – refers to the data bytes transmitted in D-PHY data lane 1
- tb.dphy_ch0.data2 – refers to the data bytes transmitted in D-PHY data lane 2
- tb.dphy_ch0.data3 – refers to the data bytes transmitted in D-PHY data lane 3

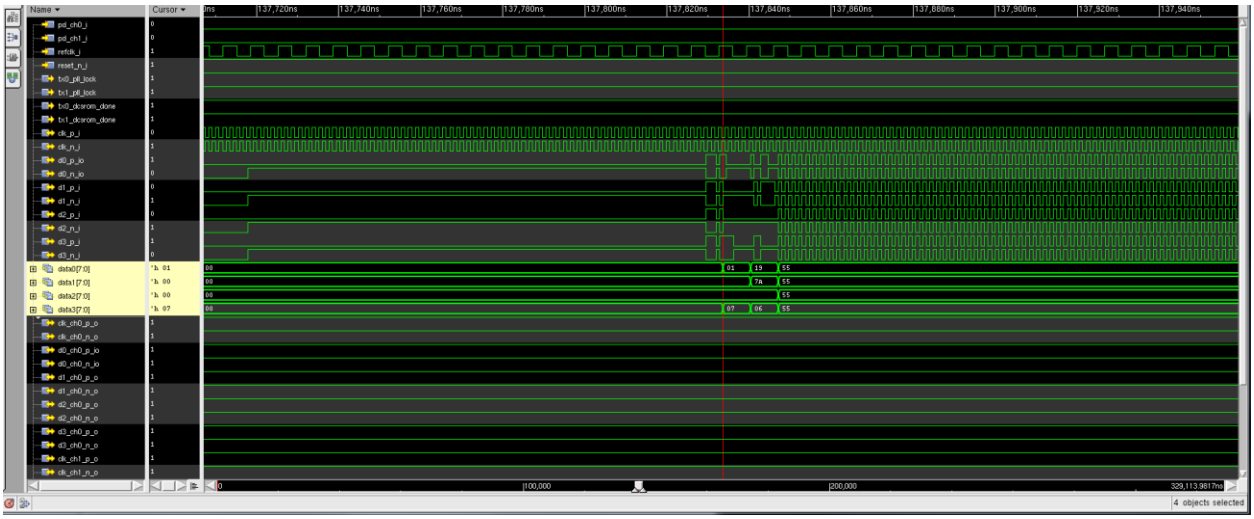

Figure 4.11. D-PHY DSI Model Video Data

4.8. Instantiating the IP

The core modules of the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP are synthesized and provided in NGO format with black box Verilog source files for synthesis. A Verilog source file, named *<instance_name>_dsi_2_dsi_ip.v*, instantiates the black box of core modules. The top-level file *<instance_name>.v* instantiates *<instance_name>_dsi_2_dsi_ip.v* .

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.

The 1:2 MIPI DSI Display Interface Bandwidth Reducer Soft IP is intended as a complete standalone solution. However, a Verilog instance template *<instancename>_inst.v* or VHDL instance template *<instancename>_inst.vhd* is also generated 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>\dsi_2_dual_dsi_br_eval\<instancename>\impl\lifmd\lse* or *\<project_dir>\dsi_2_dual_dsi_br_eval\<instancename>\impl\lifmd\synplify* directories. All required files are invoked automatically. The design can be directly synthesized, mapped, placed and routed (PAR) in the Diamond design environment after the cores are generated.

Push-button implementation of this top-level design with either Lattice Synthesis Engine (LSE) or Synopsys Synplify Pro RTL synthesis is supported via the Diamond project file *<instancename>_top.ldf* located in *\<project_dir>\dsi_2_dual_dsi_br_eval\<instancename>\impl\lifmd\<synthesis_tool>* directory.

To use the pre-built Diamond project file:

1. Choose **File** > **Open** > **Project**.
2. In the **Open Project** dialog box browse to **\<project_dir>\dsi_2_dual_dsi_br_eval\<instancename>\impl\lifmd\<synthesis_tool>**.
3. Select and open **<instancename>_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 1:2 MIPI DSI Display Interface Bandwidth Reducer IP supports Lattice's IP hardware evaluation capability, so 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 core in hardware in user-defined designs.

4.10.1. Enabling Hardware Evaluation in Diamond

If using LSE, choose **Project > Active Strategy > LSE Settings**. If using Synplify Pro, 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 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** from the menu as shown in Figure 4.12.

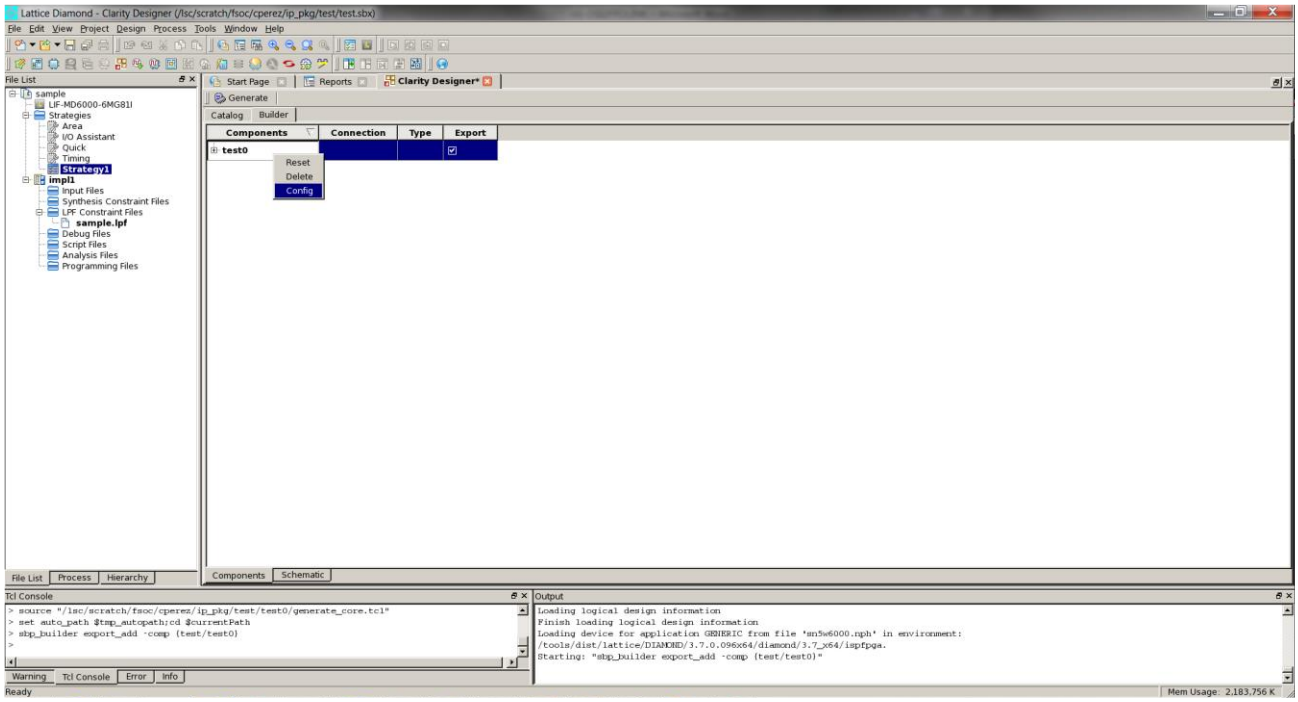

Figure 4.12. Regenerating IP in Clarity Designer

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 Display Serial Interface, version 1.1, November 22, 2011, www.mipi.org

Software documentation:

- Clarity Designer 3.8 User Manual
- Diamond 3.9.1 User Guide

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 1:2 MIPI DSI Display Interface Bandwidth Reducer 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 byte clock. The Target f_{MAX} column shows target byte clock frequency for each configuration.

Table A.1. Resource Utilization

IP User-Configurable Parameters	Slices	LUTs	Registers	sysMEM EBRs	Actual f_{MAX} (MHz)	Target f_{MAX} (MHz)
Continuous Clock Mode, Left-Right, LP Blanking	2967	5145	2642	20	86.468	75
Continuous Clock Mode, Odd-Even, LP Blanking	2967	4807	2431	4	90.310	75
Non-continuous Clock Mode, Left-Right, LP Blanking	2967	5199	2657	20	90.481	75
Non-continuous Clock Mode, Odd-Even, LP Blanking	2967	4861	2446	4	94.554	75

Appendix B. What is Not Supported

The IP does not support:

- Cycling Redundancy Check (CRC) and Error Correction Code (ECC) checking
- Bidirectional communication
- Low-level protocol error reporting
- Protocol Watchdog Timers

The 1:2 MIPI DSI Display Interface Bandwidth Reducer IP has the following design limitations:

- Minimum duration of MIPI D-PHY low-power states (tLPX) should be at least two times the byte clock period.
- Minimum byte clock is 20 MHz. Supported Rx line rate ranges from 320 Mb/s to 1200 Mb/s.
- 1-lane, 2-lane and 3-lane configurations are not supported
- Different Rx D-PHY clock mode and Tx D-PHY clock mode settings are not guaranteed. This option is disabled.
- Left-Right mode and Odd-Even mode can only support up to 16384 Rx word count.
- Design can only support RGB888 data type.
- Mismatch between Rx and Tx line times causes overflow/underflow inside line buffers. For left-right mode, Rx blanking period must be at least as long as Tx blanking period. For odd-even mode, Rx blanking must be exactly the same as Tx blanking period. Tx blanking period is based on minimum D-PHY timing requirements and depends on the byte clock frequency.
- HS blanking feature can be used for shorter blanking periods which cannot be supported when Tx is using LP blanking. The HS blanking parameters (HSA, HBP and HFP) must be computed based on the Rx HSA, HBP and HFP.
- Only non-burst DSI video modes are supported. If HS blanking feature is used, only non-burst with sync pulses video mode is supported.
- If high-speed DCS is selected, only one DCS ROM is used. DCS for Tx channel 0 is transmitted first, followed by DCS for Tx channel 1.
- When enabling debug signals, only six internal signals can be probed at a time due to device limitations. Enabling all miscellaneous signals causes place and route issues.
- HS trail should be minimum of 3 byte clock cycles.
- When using HS blanking feature, we recommend to use HS blanking also in Rx side.
- When using LP blanking feature, we recommend to use LP blanking also in Rx side.

Appendix C. Initializing the DCS ROM

Display Command Set (DCS) initialization is used to configure the command registers of a DSI-compliant display. The bridge has an option to perform this in high-speed or in low-power mode.

In either DCS mode, the number of entries must correspond to the Number of DCS Words indicated in the interface.

There should be no empty lines within the text file. Comments within the file are not supported.

Low-Power Mode

To initialize the DCS ROM in low-power mode, the input file must contain one byte of data in each line, in hex format. Figure C.1. shows the sample entries.

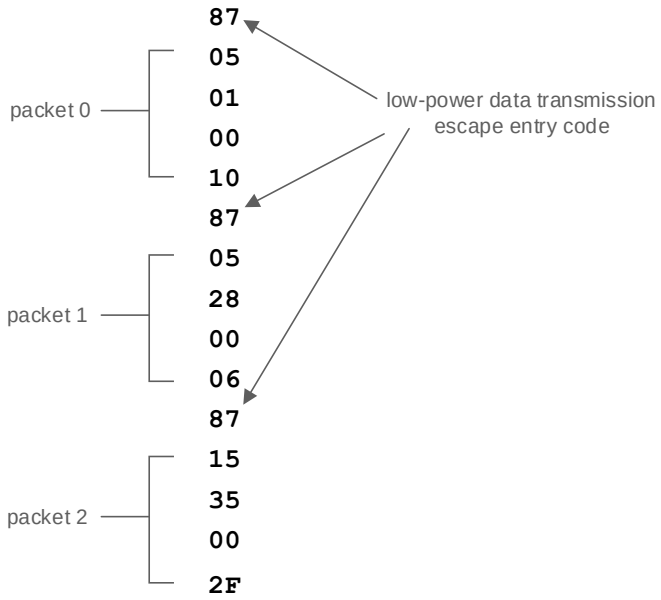

Figure C.1. DCS ROM for DCS Low-Power Mode

The 8'h87 byte indicates the start of a new packet. In this example, the DCS Controller breaks down the DCS words into 3 packets. The last entry should be the last valid byte. DCS Word Count in this example is 15.

High-Speed Mode

When the DCS ROM initialization is in high-speed mode, the interval between high-speed transmissions may be set through the DCS ROM Wait Time parameter. Multiple packets may be concatenated to reduce overhead of frequent switching between low-power state and high-speed mode.

The entries within the input file should be in the following format:

<trail bit indicator><DCS byte lane3><DCS byte lane2><DCS byte lane1><DCS byte lane0>

For each high-speed transmission, each lane must start with the SoT pattern 8'hB8, and the last word should be made up of complete trail bytes with the trail indicator bit set to 1. The design checks this trail bit indicator to determine the end of the high-speed transmission.

Sample DCS for Gear 8

Figure C.2 shows the sample entries for the DCS initialization file of a 4-lane, gear 8 configuration.

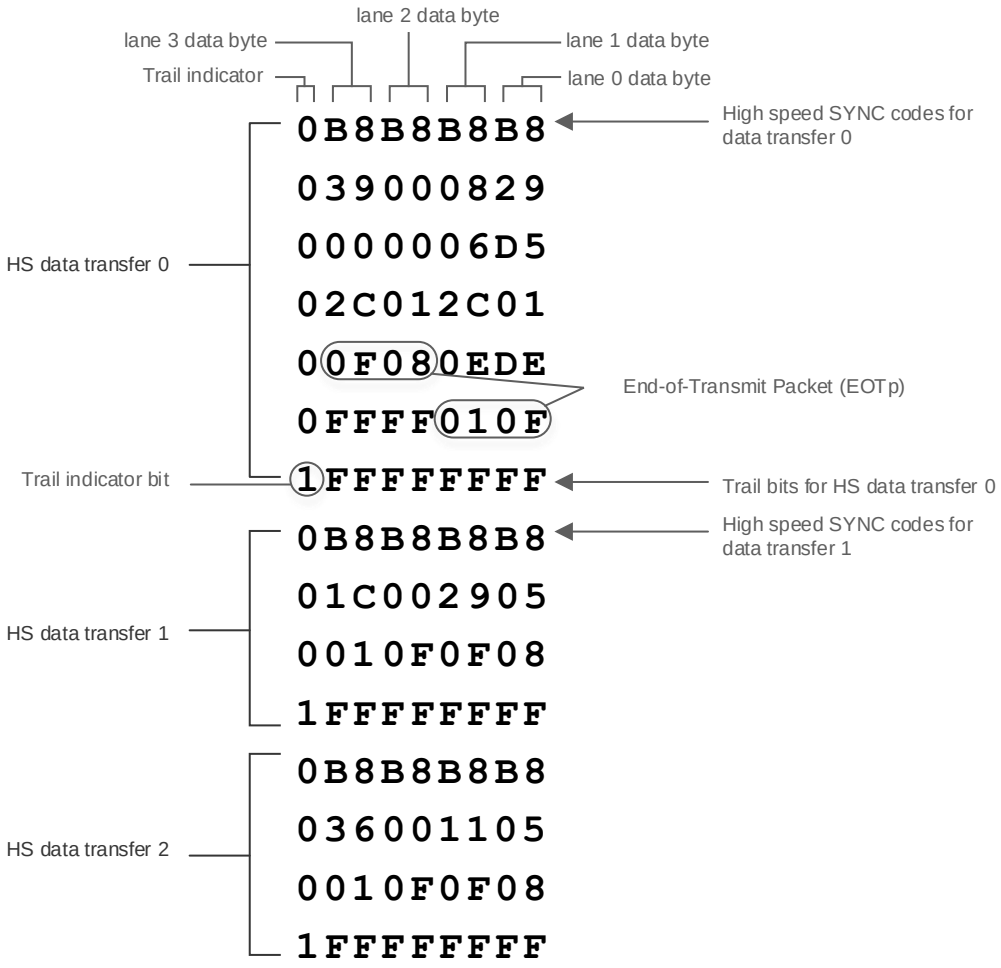

Figure C.2. Sample DCS ROM for x4 Gear 8 DCS High-Speed Mode

In this example, an End-of-Transmit packet is sent after each DCS packet. The last word after each high-speed transmission must be made up completely of trail bits. The DCS Word Count in this example is 15.

Sample DCS for Gear 16

The byte order of DCS words for gear 16 configuration should follow the order described in Figure 1.1.

Figure C.3 shows the sample entries for 4-lane, gear 16 configuration.

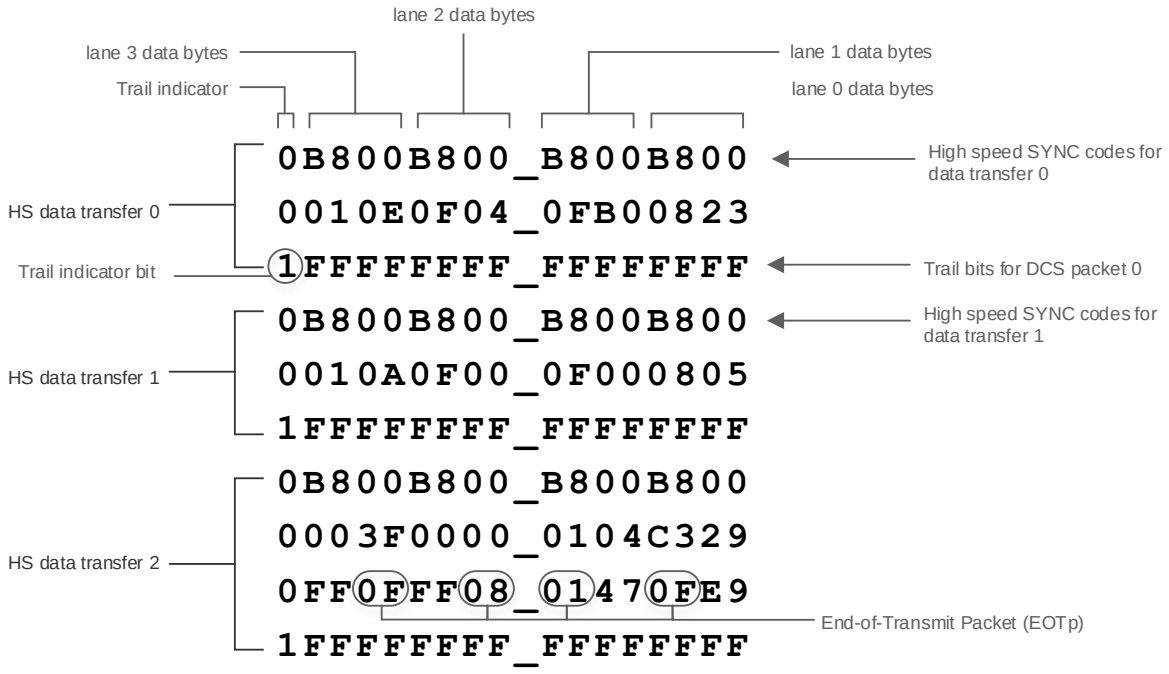

Figure C.3. Sample DCS ROM for x4 Gear 16 DCS High-Speed Mode

This example contains three DCS packets concatenated with EoTP at the end of each packet. The first word for each lane contains the MIPI D-PHY high-speed synchronization sequence 8'hB8, padded with zeros at the start. The zero padding is used for alignment purposes only. These may be removed, but the data bytes should be adjusted accordingly. The DCS Word Count in this example is 10.

Sample DCS ROM files are also available in the *dsi_to_dsi_v<x>.<y>/samples* folder in the IP installation directory.

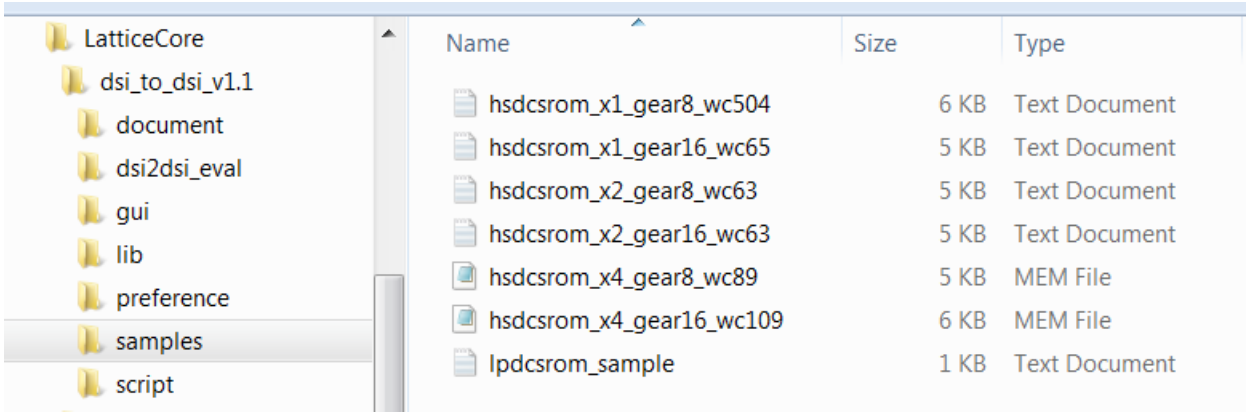

Figure C.4. Directory Containing the Sample DCS ROM Initialization Files

Appendix D. HS Blanking Requirements

By default, the 1:2 MIPI DSI Display Interface Bandwidth Reducer IP transmits low-power blanking during a blanking period. The duration of the low-power blanking period is based on minimum MIPI D-PHY timing requirements (T_{LPX}, $T_{HS-PREPARE}$, $T_{HS-ZERO}$, and so on), and depends on the byte clock frequency. It is difficult to meet short and timing critical blanking requirements with low-power blanking.

High-Speed (HS) blanking feature is available to support shorter blanking periods that cannot be supported with low-power blanking. The high-speed blanking feature covers Horizontal Sync Active (HSA), Horizontal Back Porch (HBP) and Horizontal Front Porch (HFP) inside Vertical Active (VACT) region.

HS blanking feature is also recommended for Odd-Even mode, where it is critical to match the Rx and Tx line times to avoid FIFO overflow/underflow inside the line buffers. To properly match Rx and Tx line times, the parameters must be computed as described below.

Horizontal Sync Active (HSA)

The HSA parameter specifies the effective word count of the blanking packet transmitted between HSYNC start and HSYNC end short packets. It is described as the effective word count because HSA is transmitted as two consecutive blanking packets with word counts that add up to HSA - 6. The remaining 6 bytes in HSA are used for the 4-byte header and 2-byte footer of the second blanking packet. The minimum HSA is 26 due to design limitation.

*HSA = (((Rx HSA + 6) / 8) - 1) * 4 - 2*

Horizontal Back Porch (HBP)

The HBP parameter specifies the effective word count of the blanking packet transmitted between HSYNC end short packet and active data long packet. It is described as the effective word count because HBP is transmitted as two consecutive blanking packets with word counts that add up to HBP - 6. The remaining 6 bytes in HSA are used for the 4-byte header and 2-byte footer of the second blanking packet. The minimum HBP is 38 due to design limitation.

*HBP = (((Rx HBP + 6) / 8) - 1) * 4 - 2*

Horizontal Front Porch (HFP)

The HFP parameter specifies the word count of the blanking packet transmitted between active data long packet and HSYNC start short packet. The minimum HFP is 28 due to design limitation.

*HFP = (((Rx HFP + 4) / 8) - 3) * 4*

Revision History

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