1 to N MIPI CSI-2/DSI Duplicator with CrossLink-NX

Reference Design

FPGA-RD-02215-1.2

March 2025

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor's inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice's inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AP	Application Processor
CRC	Cyclic Redundancy Check
CSI-2	Camera Serial Interface 2
DDR	Double Data Rate
DSI	Display Serial Interface
EBR	Embedded Block RAM
ECC	Error Correction Code
FS	Frame Start
HS	High Speed
ID	Identification Data
LP	Low Power
MIPI	Mobile Industry Processor Interface
PLL	Phase Locked Loop
GPLL	General Purpose PLL
RX	Receiver
STA	Static Timing Analysis
TX	Transmitter
VC	Virtual Channel
VSS	Vsync Start
WC	Word Count

1. Introduction

Many new applications such as virtual reality, augmented reality, and digital cameras require expansion of the number of camera or display interface on application processors (AP). This often occurs when there is either not enough ports or some ports are used for other purposes on the AP.

The Lattice Semiconductor 1 to-N MIPI® CSI-2/DSI Duplicator with CrossLink-NX reference design for CrossLink™-NX devices has one to four-channel outputs for duplicator. CrossLink-NX has two MIPI Hardened macro IPs, which can be used as MIPI TX or RX module (D-PHY Hardened IP). The TX and RX modules can also be realized by a soft macro utilizing general DDR modules (D-PHY Soft IP).

1.1. Supported Device and IP

Table 1.1 indicates the device and compatible IP versions supported in this reference design.

Table 1.1. Supported Device and IP

Device Family	Part Number	Compatible IP
CrossLink-NX	LIFCL-40	D-PHY Receiver IP version 1.9.0 D-PHY Transmitter IP version 2.2.0

1.2. Features

- MIPI CSI-2 or DSI stream on one RX channel is duplicated and sent out on one to four TX channels.
- RX channel can have one, two, or four lanes.
- Using Hardened D-PHY IP on Rx channel is recommended to save FPGA resources.
- Maximum RX bandwidth is 2.5 Gbps per lane using D-PHY Hardened IP and 1.5 Gbps per lane using D-PHY Soft IP.
- Number of TX lanes can be one, two, or four. This is independent from the number of RX lanes.
- Maximum TX bandwidth is 2.5 Gbps per lane using D-PHY Hardened IP and 1.5 Gbps per lane using D-PHY Soft IP.
- Non-continuous clock mode on RX channels is possible as long as the continuous clock is obtained internally or fed directly from the pin.
- Non-continuous or continuous clock mode is possible for TX channels.

1.3. Block Diagram and Clocking Scheme

Figure 1.1 shows the block level diagram of the 1 to N MIPI CSI-2/DSI Duplicator with CrossLink-NX reference design with four TX channels.

Instead of using individual PLL in D-PHY TX IP, it is possible to use on-chip GPLL to generate a required high-speed clock and feed it to multiple D-PHY TX IP.

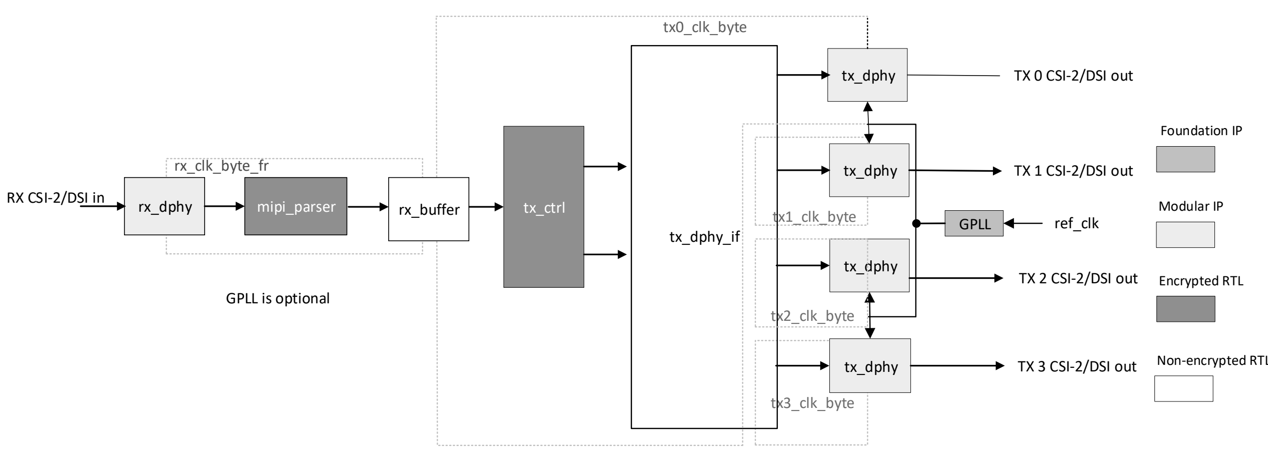

Figure 1.1. 1 to N MIPI CSI-2/DSI Duplicator Block Diagram

Clocking scheme mainly depends on RX clock mode and TX PLL mode. In the case of Hardened D-PHY TX, internal PLL mode means using the internal PLL equipped with MIPI Hardened macro IP, which is different from on-chip GPLL. In the case of Soft D-PHY TX, on-chip GPLL is assigned to provide a high-speed clock. The only difference is that GPLL is automatically instantiated inside the IP or user-defined GPLL is used to generate the high-speed clock. In general, external PLL mode is preferred when multiple Soft D-PHY IPs are used on TX channels.

Figure 1.2 shows a clocking scheme example in RX continuous clock mode for two TX channels with internal PLL configuration. sync_clk is required only in case that RX D-PHY is Soft IP. In this case, a high-speed clock is generated by PLL in each TX D-PHY.

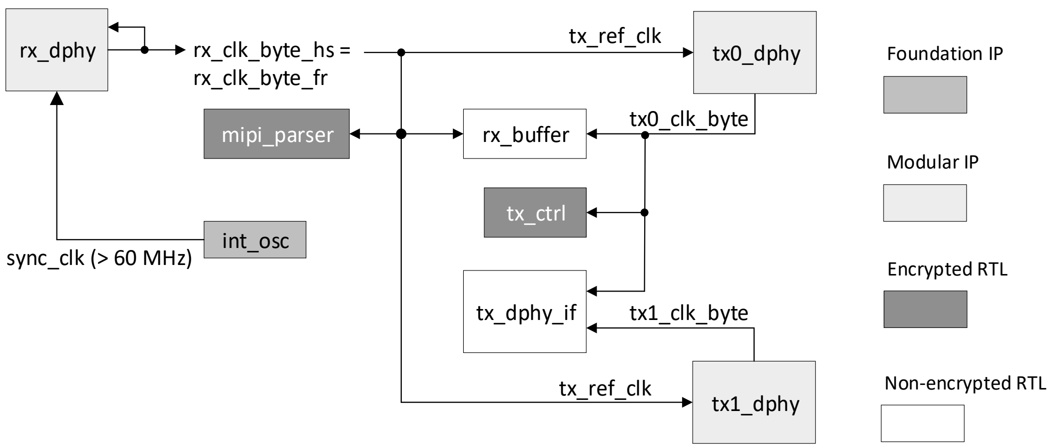

Figure 1.2. Clocking Scheme Example in RX Continuous Clock Mode for 2-Channel TX with Internal PLL Mode

Figure 1.3 shows a clocking scheme example in RX non-continuous clock mode for two TX channels with internal PLL configuration. An external clock is required to generate continuous RX byte clock. sync_clk is required only in case that RX D-PHY is Soft IP and this clock can be shared with rx_clk_byte_fr when rx_clk_byte_fr is over 60 MHz. In this case, a high-speed clock is generated by PLL in each TX D-PHY.

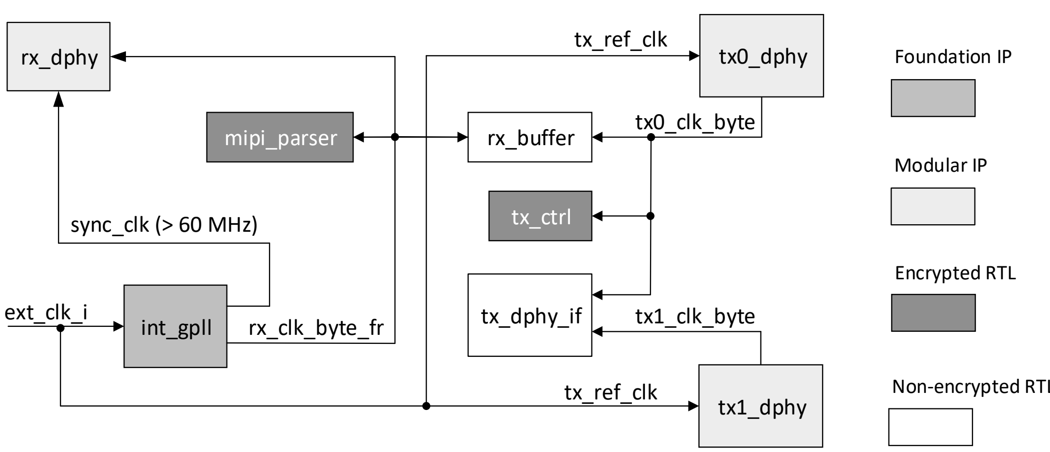

Figure 1.3. Clocking Scheme Example in RX Non-Continuous Clock Mode for 2-Channel TX with Internal PLL Mode

Figure 1.4 shows a clocking scheme example in RX continuous clock mode for two TX channels with external PLL configuration. sync_clk is required only in case that RX D-PHY is Soft IP. In this case, GPLL is used to generate high-speed clock used in D-PHY TX. This clock must be the same speed as TX lane bandwidth for Hardened D-PHY TX and a half speed for Soft D-PHY TX. For Soft D-PHY TX, the high-speed clock txN_pll2_clk has the same frequency as txN_pll_clk but the clock is 90-degree phase shifted from txN_pll_clk. The int_gpll needs to generate two different high-speed clocks when both Hardened and Soft D-PHY IP are used on TX channels with external PLL mode. In other words, tx0_pll_clk, tx1_pll_clk, tx0_pll2_clk, and tx1_pll2_clk can be shared when tx0_dphy and tx1_dphy are same D-PHY type (Hardened D-PHY or Soft D-PHY). tx_ref_clk is not required for Hardened D-PHY TX. As the maximum clock frequency on GPLL output clock is 800 MHz, the maximum lane bandwidth for Hardened D-PHY TX with external PLL mode is 800 Mbps. Soft D-PHY TX can operate up to 1500 Mbps regardless of internal or external PLL mode.

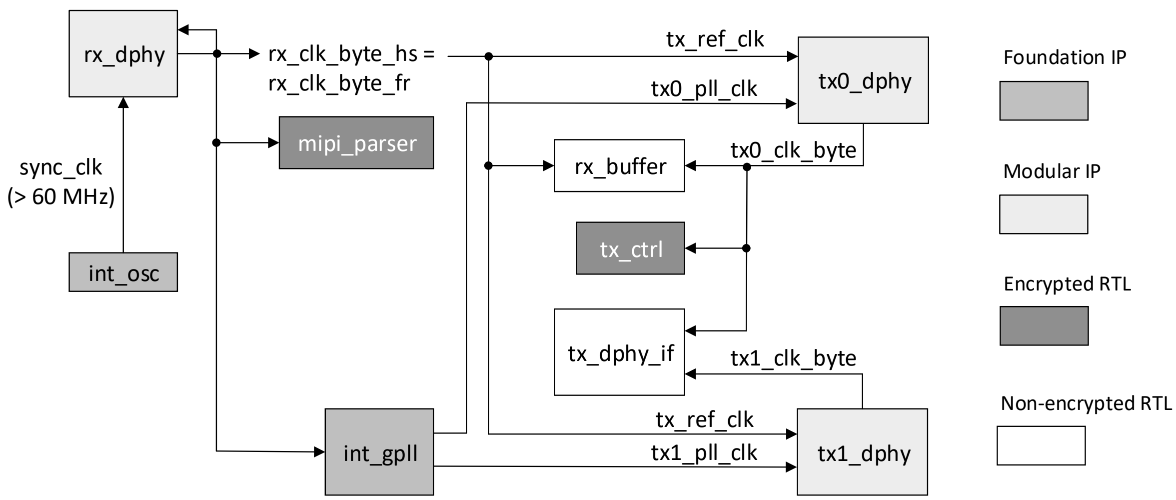

Figure 1.4. Clocking Scheme Example in RX Continuous Clock Mode for 2-Channel TX with External PLL Mode

Figure 1.5 shows a clocking scheme example in RX non-continuous clock mode for to two TX channel with external PLL configuration. An external clock is required to generate continuous RX byte clock and high-speed clocks for D-PHY TX. sync_clk is required only in case that RX D-PHY is Soft IP and this clock can be shared with rx_clk_byte_fr when rx_clk_byte_fr is over 60 MHz. tx0_pll_clk, tx1_pll_clk, tx0_pll2_clk, and tx1_pll2_clk can be shared when tx0_dphy and tx1_dphy are same D-PHY type (Hardened D-PHY or Soft D-PHY).

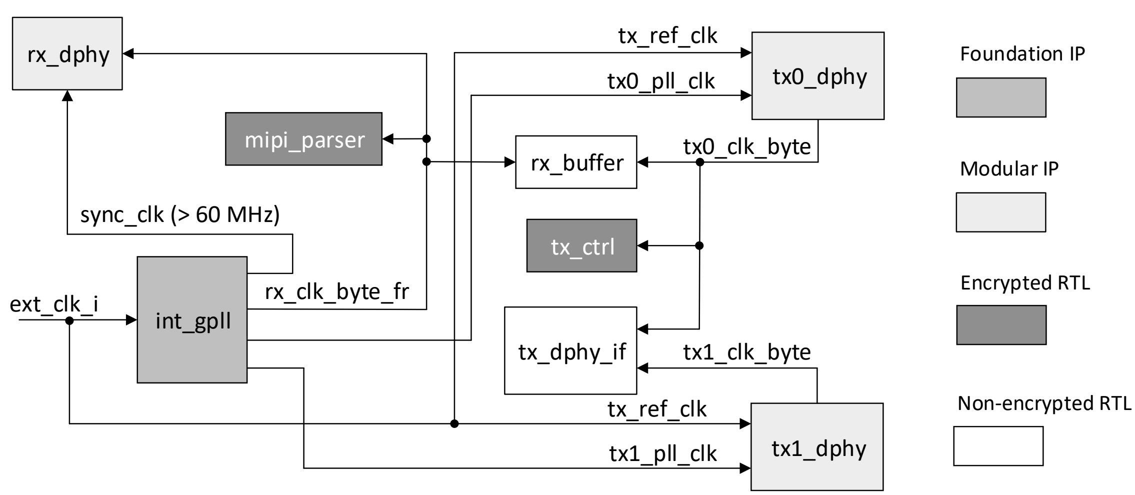

Figure 1.5. Clocking Scheme Example in RX Non-Continuous Clock Mode for 2-Channel TX with External PLL Mode

Table 1.2 – Table 1.5 show typical clock resources for different configurations. These tables are based on the concept that two Hardened D-PHY IPs are always used. This leaves more logic resources for other purposes and reduces power consumption. For example, using only Hardened D-PHYs is assumed in the case of 1:1 duplication. GPLL output GPLL_clkop is assigned to the highest frequency, GPLL_clkos is assigned to the second, and so on. These are guidelines and you can change the clocking scheme as long as the design is functional.

Table 1.2. Clock Resources in Case of RX Hardened D-PHY with Continuous Clock Mode

RX D-PHY Related	RX_CLK_MODE	HS_ONLY (Continuous Clock)				
	D-PHY IP Type	Hardened				
	max. RX Lane BW	2500 Mbps				
	clk_lp_ctrl_i	NA				
	sync_clk_i	NA				
	rx_clk_byte_fr	From RX D-PHY				
Use ext_clk_i		No				
Use off-IP GPLL		No	Yes			
GPLL ref_clk_i		NA	rx_clk_byte_fr		rx_clk_byte_fr	
TX D-PHY Related	TX D-PHY Type	Hardened only	Hardened: Internal, Soft: External		Hardened and Soft: External	
	max. TX Lane BW	2500 Mbps	1500 Mbps		800 Mbps	
	TX D-PHY IP	Hardened	Hardened	Soft	Hardened	Soft
	TX PLL mode	Internal	Internal	External	External	External
	tx_ref_clk	rx_clk_byte_fr	rx_clk_byte_fr	rx_clk_byte_fr	NA	rx_clk_byte_fr
	tx_pll_clk	NA	NA	GPLL_clkop	GPLL_clkop	GPLL_clkos
Suitable channel configuration		1:1	1:2, 1:3, 1:4		1:3, 1:4	

Table 1.3. Clock Resources in Case of RX Soft D-PHY with Continuous Clock Mode

RX D-PHY Related	RX_CLK_MODE	HS_ONLY (Continuous Clock)				
	D-PHY IP Type	Soft				
	max. RX Lane BW	1500 Mbps				
	clk_lp_ctrl_i	NA				
	sync_clk_i	OSC				
	rx_clk_byte_fr	From RX D-PHY				
Use ext_clk_i		No				
Use off-IP GPLL		No	Yes			
GPLL ref_clk_i		NA	rx_clk_byte_fr		rx_clk_byte_fr	
TX D-PHY Related	TX D-PHY Type	Hardened only	Hardened: Internal, Soft: External		Hardened and Soft: External	
	max. TX Lane BW	2500 Mbps	1500 Mbps		800 Mbps	
	TX D-PHY IP	Hardened	Hardened	Soft	Hardened	Soft
	TX PLL mode	Internal	Internal	External	External	External
	tx_ref_clk	rx_clk_byte_fr	rx_clk_byte_fr	rx_clk_byte_fr	NA	rx_clk_byte_fr
	tx_pll_clk	NA	NA	GPLL_clkop	GPLL_clkop	GPLL_clkos
Suitable channel configuration		1:2	1:3, 1:4		1:3, 1:4	

Table 1.4. Clock Resources in Case of RX Hardened D-PHY with Non-Continuous Clock Mode

RX D-PHY Related	RX_CLK_MODE	HS_LP (Non-continuous Clock)				
	D-PHY IP Type	Hardened				
	max. RX Lane BW	2500 Mbps				
	clk_lp_ctrl_i	ext_clk_i				
	sync_clk_i	NA				
	rx_clk_byte_fr	GPLL_clkos2				
Use ext_clk_i		Yes				
Use off-IP GPLL		Yes				
GPLL ref_clk_i		ext_clk_i				
TX D-PHY Related	TX D-PHY Type	Hardened only	Hardened: Internal, Soft: External		Hardened and Soft: External	
	max. TX Lane BW	2500 Mbps	1500 Mbps		800 Mbps	
	TX D-PHY IP	Hardened	Hardened	Soft	Hardened	Soft
	TX PLL mode	Internal	Internal	External	External	External
	tx_ref_clk	ext_clk_i	ext_clk_i	ext_clk_i	NA	ext_clk_i
	tx_pll_clk	NA	NA	GPLL_clkop	GPLL_clkop	GPLL_clkos
Suitable channel configuration		1:1	1:2, 1:3, 1:4		1:2, 1:3, 1:4	

Table 1.5. Clock Resources in Case of RX Soft D-PHY with Non-Continuous Clock Mode

RX D-PHY Related	RX_CLK_MODE	HS_LP (Non-continuous Clock)				
	D-PHY IP Type	Soft				
	max. RX Lane BW	1500 Mbps				
	clk_lp_ctrl_i	ext_clk_i				
	sync_clk_i	OSC				
	rx_clk_byte_fr	GPLL_clkos2				
Use ext_clk_i		Yes				
Use off-IP GPLL		Yes				
GPLL ref_clk_i		ext_clk_i				
TX D-PHY Related	TX D-PHY Type	Hardened only	Hardened: Internal, Soft: External		Hardened and Soft: External	
	max. TX Lane BW	2500 Mbps	1500 Mbps		800 Mbps	
	TX D-PHY IP	Hardened	Hardened	Soft	Hardened	Soft
	TX PLL mode	Internal	Internal	External	External	External
	tx_ref_clk	ext_clk_i	ext_clk_i	ext_clk_i	NA	ext_clk_i
	tx_pll_clk	NA	NA	GPLL_clkop	GPLL_clkop	GPLL_clkos
Suitable channel configuration		1:2	1:3, 1:4		1:3, 1:4	

1.4. RX/TX Permutations

Table 1.6 shows the possible RX/TX permutations and Figure 1.6 shows the calculated bandwidth in the Excel sheet provided with this reference design. Some unrealistic permutations are excluded due to bandwidth limitations. When the lane bandwidth is around 1500 Mbps, either Gear 8 or Gear 16 may be used in case of Hardened D-PHY. If Gear 8 cannot meet timings (see static timing analysis (STA) report), Gear 16 should be used. Soft D-PHY can support only Gear 8. The maximum lane bandwidth depends on the type of D-PHY IP, Hardened D-PHY or Soft D-PHY. In case of two-channel or more outputs, at least one Soft D-PHY is used on either RX or TX channel so that maximum lane bandwidth is 1500 Mbps on that channel. Additionally, when the external PLL mode is used in Hardened D-PHY TX, the maximum TX lane bandwidth is limited to 800 Mbps due to GPLL capability. The permutations shown in Table 1.6 is based on the assumption that total bandwidth (lane bandwidth x number of lane) is equal between RX and one TX channel. In some cases, RX and TX total bandwidth could be different and the values shown in the table and figure might not directly apply. These cases are described in rx_buffer section.

Table 1.6. RX/TX Permutations

Number of RX Lane	RX Gear[2]	RX Lane Bandwidth[1] x (Mbps)	Number of TX Lane	TX Gear[2]	TX Lane Bandwidth[1] y (Mbps)	TX Byte Clock/ RX Byte Clock
1	16	~1500 ≤ x ≤ 2500	1	16	~1500 ≤ y ≤ 2500	1
			2	8	~750 ≤ y ≤ 1250	1
			4	8	~375 ≤ y ≤ 625	0.5
	8	160 ≤ x ≤ ~1500	1	8	160 ≤ y ≤ ~1500	1
		320 ≤ x ≤ ~1500	2		160 ≤ y ≤ ~750	0.5
		640 ≤ x ≤ ~1500	4		160 ≤ y ≤ ~375	0.25
2	16	~1500 ≤ x ≤ 2500	1	16	y ≥ ~3000	NA
		~1500 ≤ x ≤ 2500	2	8	y ≥ ~1500	NA
		~1500 ≤ x ≤ 2500	4	8	~750 ≤ y ≤ 1250	1
	8	~750 ≤ x ≤ 1250	1	16	~1500 ≤ y ≤ 2500	1
		160 ≤ x ≤ ~750		8	320 ≤ y ≤ ~1500	2
		160 ≤ x ≤ ~1500	2	8	160 ≤ y ≤ ~1500	1
		320 ≤ x ≤ ~1500	4	8	160 ≤ y ≤ ~750	0.5
4	16	~1500 ≤ x ≤ 2500	1	16	y ≥ ~6000	NA
		~1500 ≤ x ≤ 2500	2	16	y ≥ ~3000	NA
		~1500 ≤ x ≤ 2500	4	16	~1500 ≤ y ≤ 2500	1
	8	~375 ≤ x ≤ 625	1	16	~1500 ≤ y ≤ 2500	2
		160 ≤ x ≤ ~375		8	640 ≤ y ≤ ~1500	4
		~750 ≤ x ≤ 1250	2	16	~1500 ≤ y ≤ 2500	1
		160 ≤ x ≤ ~750		8	320 ≤ y ≤ ~1500	2
		160 ≤ x ≤ ~1500	4	8	160 ≤ y ≤ ~1500	1

Notes:

1. The maximum lane bandwidth depends on the type of D-PHY IP (2500 Mbps: Hardened D-PHY, 1500 Mbps: Soft D-PHY). Since CrossLink-NX has two Hardened D-PHYs, the maximum TX lane bandwidth is always 1500 Mbps for 3 and 4 TX channel configurations.
2. The bandwidth border guideline of Gear selection between 8 and 16 is 1500 Mbps. When the lane bandwidth is around 1500 Mbps, you can select either one based on the result of the static timing report (STA). Note that Gear 16 is available for only Hardened D-PHY. Soft D-PHY supports only Gear 8.

1 to N MIPI CSI-2/DSI Duplicator with CrossLink-NX Parameter Calculator			
RX D-PHY IP Type	Soft		
Number of RX Lanes	1		
RX Gear	8		**Gear 8 only**
RX Line Rate (per lane)	600	Mbps	**up to 1500**
RX Clock Mode	Non-cont.		
RX DPHY Clock Frequency	300	MHz	
RX Byte Clock Frequency	75	MHz	
Number of TX Channels	2		
Number of TX Lanes	1		
TX Gear	8		**Gear 8 or 16**
TX Line Rate (per lane)	600	Mbps	**up to 2500**
Hard D-PHY TX PLL Mode	internal		
TX DPHY Clock Frequency	300	MHz	
TX Byte Clock Frequency	75	MHz	
Soft D-PHY TX should be in external PLL mode			
assuming to always use two Hard D-PHY IP			

Set by user

Figure 1.6. Bandwidth and Parameter Calculator

The excel file for the parameter calculator is located to the design package source code. To locate *mipi_dup_RD_NX.xlsx*, open the source code folder and locate the folder *doc*.

2. Parameters and Port List

There are two directive files for this reference design:

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

You can modify these directives according to your own configuration. The settings in these files must match RX D-PHY IP, TX D-PHY IP, and other module settings described in the Design and Module Description section.

2.1. Synthesis Directives

Table 2.1 shows the synthesis directives that affect this reference design. These are used for both synthesis and simulation. As shown in Table 2.1 and Table 2.2, some parameter selections are restricted by other parameter settings.

Table 2.1. Synthesis Directives

Category	Directive	Remarks
RX D-PHY Type	CSI2	Only one of these two directives must be defined.
	DSI	
RX channel lane count	NUM_RX_LANE_1	Number of lanes in each RX channel. Only one of these three directives must be defined.
	NUM_RX_LANE_2	
	NUM_RX_LANE_4	
RX D-PHY Variant	RX_DPHY_HARD	Only one of these two directives must be defined. When not defined RX_DPHY_HARD, RX channel uses Soft D-PHY.
	RX_DPHY_SOFT	
RX DPHY Clock Gear	RX_GEAR_8	Only one of these directives must be selected. Gear 16 can be used for only Hardened D-PHY.
	RX_GEAR_16	
RX D-PHY Clock Mode[1, 2]	RX_CLK_MODE_HS_ONLY	RX D-PHY Clock mode. Only one of these two directives must be defined.
	RX_CLK_MODE_HS_LP	
RX DPHY FIFO	RX DPHY_FIFO_MISC_ON	Enable Rx DPHY FIFO Miscellaneous signals. Define this only if RX FIFO is enabled and RX FIFO Misc signals option is checked in the RX DPHY HARD/SOFT user interface.
Wait for FS/VSS	WAIT_FS	Wait for FS (Frame Start) or VSS (Vsync Start) to begin capturing RX data. Capturing begins with any data if not defined.
RX Buffer Depth	RX_BUFFER_DEPTH_512	Depth of RX Buffer FIFO. Only one of these four directives must be defined. 4096 cannot be defined in case of (NUM_RX_LANE_4 and RX_GEAR_16) due to the limitation of available EBR.
	RX_BUFFER_DEPTH_1024	
	RX_BUFFER_DEPTH_2048	
	RX_BUFFER_DEPTH_4096	
Byte Data Read Delay	BD_RD_DLY {value}	Byte Data Read Delay from RX Buffer in rx_byte_clk cycles. The value must be 1 – 8191.
TX Channel count	NUM_TX_CH_1	Number of TX channel. Only one of these four directives must be defined.
	NUM_TX_CH_2	
	NUM_TX_CH_3	
	NUM_TX_CH_4	
TX channel lane count	NUM_TX_LANE_1	Number of lanes in TX channel. Only one of these three directives must be defined.
	NUM_TX_LANE_2	
	NUM_TX_LANE_4	
TX D-PHY Clock Gear	TX_GEAR_8	TX D-PHY Clock Gear. Only one of these two directives must be defined. Gear 16 can be used for only Hardened D-PHY.
	TX_GEAR_16	
TX0 Hardened D-PHY	TX0_DPHY_HARD	Enable Hardened D-PHY on TX channel #0. When not defined, TX channel #0 uses Soft D-PHY.
TX1 Hardened D-PHY	TX1_DPHY_HARD	Enable Hardened D-PHY on TX channel #1. When not defined, TX channel #1 uses Soft D-PHY.

Category	Directive	Remarks
TX2 Hardened D-PHY	TX2_DPHY_HARD	Enable Hardened D-PHY on TX channel #2. When not defined, TX channel #2 uses Soft D-PHY.
TX3 Hardened D-PHY	TX3_DPHY_HARD	Enable Hardened D-PHY on TX channel #3. When not defined, TX channel #3 uses Soft D-PHY.
TX0 External PLL[3]	TX0_EXT_PLL	Hardened D-PHY: Use GPLL to receive high-speed clock then routed to DPHY TX PLL. When not defined, it goes directly to the DPHY TX channel #0 PLL. Soft D-PHY: Use GPLL regardless of this directive. When defined, GPLL is instantiated within D-PHY TX IP and a generated clock by GPLL cannot be shared with other modules.
TX1 External PLL[3]	TX1_EXT_PLL	Hardened D-PHY: Use GPLL to receive high-speed clock then routed to DPHY TX PLL. When not defined, it goes directly to the DPHY TX channel #1 PLL. Soft D-PHY: Use GPLL regardless of this directive. When defined, GPLL is instantiated within D-PHY TX IP and a generated clock by GPLL cannot be shared with other modules.
TX2 External PLL[3]	TX2_EXT_PLL	Hardened D-PHY: Use GPLL to receive high-speed clock then routed to DPHY TX PLL. When not defined, it goes directly to the DPHY TX channel #2 PLL. Soft D-PHY: Use GPLL regardless of this directive. When defined, GPLL is instantiated within D-PHY TX IP and a generated clock by GPLL cannot be shared with other modules.
TX3 External PLL[3]	TX3_EXT_PLL	Hardened D-PHY: Use GPLL to receive high-speed clock then routed to DPHY TX PLL. When not defined, it goes directly to the DPHY TX channel #3 PLL. Soft D-PHY: Use GPLL regardless of this directive. When defined, GPLL is instantiated within D-PHY TX IP and a generated clock by GPLL cannot be shared with other modules.
TX D-PHY Clock Mode[1]	TX_CLK_MODE_HS_ONLY TX_CLK_MODE_HS_LP	TX D-PHY Clock mode. Only one of these two directives must be defined.
Keep HS mode	KEEP_HS	Keep the clock lane in HS mode during the horizontal blanking periods of active video lines when defined. Effective when CSI2 and TX_CLK_MODE_HS_LP are defined.

Notes:

1. HS_LP mode means non-continuous clock mode and HS_ONLY means continuous clock mode.
2. HS_LP mode works only if RX byte clock for corresponding RX channel can be generated internally or directly fed from I/O pin.
3. SOFT DPHY TX IP always uses external PLL mode.

2.2. Simulation Directives

Table 2.2 shows the simulation directives for this reference design.

Table 2.2. Simulation Directives

Category	Directive	Remarks
simulation	SIM	Select behavioral models for simulation.
DSI Sync Mode	DSI_SYNC_MODE_EVENT	Select non-burst with sync event or no-burst with sync pulse mode. Applicable when DSI is defined in synthesis_directives.v.
	DSI_SYNC_MODE_PULSE	
EoTp insertion	EOTP	Insert EoTp (End of Transmission Packet) when defined. Applicable when DSI is defined in synthesis_directives.v.
VSA length	VSA_LENGTH {value}	Number of Vsync active lines. Applicable when DSI is defined in synthesis_directives.v.
VBP length	VBP_LENGTH {value}	Number of Vertical Back Porch lines. Applicable when DSI is defined in synthesis_directives.v.
VFP length	VFP_LENGTH {value}	Number of Vertical Front Porch lines. Applicable when DSI is defined in synthesis_directives.v.
HSA length	HSA_LENGTH {value}	Number of WC in Null Packet of Hsync active period. Applicable when DSI is defined in synthesis_directives.v.
HBP length	HBP_LENGTH {value}	Number of WC in Blanking Packet of Horizontal Back Porch. Applicable when DSI is defined in synthesis_directives.v.
HFP length	HFP_LENGTH {value}	Number of WC in Blanking Packet of Horizontal Front Porch. Applicable when DSI is defined in synthesis_directives.v.
Reference clock period	REF_CLK_PERIOD {value}	Reference clock period in ps
RX DPHY clock period	RX_DPHY_CLK_PERIOD {value}	RX Channel DPHY clock period in ps
RX DPHY Byte clock period	RX_FREQ_TGT	RX DPHY byte clock in ps
TX DPHY Byte clock period	TX_FREQ_TGT	TX DPHY byte clock in ps
TX DPHY clock period	TX_DPHY_CLK_PERIOD {value}	TX Channel DPHY clock period in ps
TX Wait less	TX_WAIT_LESS_15MS	—
RX Virtual channel	RX_VC {value}	Virtual channel ID of the packet. Value is 0.
Initial delay on RX channel	RX_DELAY {value}	Initial delay to activate RX Channel in ps
Gap (LP) time between active lines on RX Channel	RX_DPHY_LPS_GAP {value}	Gap time on RX Channels in ps
Gap (LP) time between Frame End and Frame Start on RX Channel	RX_DPHY_FRAME_GAP {value}	Gap time on RX Channels in ps
Video data configuration on RX Channel*	NUM_FRAMES {value}	Number of frames to feed.
	NUM_LINES {value}	Number of active lines per frame.
	NUM_PIXELS {value}	Number of pixels per active line.
	RAW8	Data Type of the payload video data. Only one among RAW8, RAW10, RAW12 and RGB888 must be defined.
	RAW10	
	RAW12	
	RGB888	
Initialization Time monitor	MISC_TINITDONE	Enables internal signal tx0_tinit_done monitored by the testbench. Enable this directive when "Bypass tINIT counter" is unchecked in TX D-PHY settings in Clarity for tx_dphy.

***Note**: RGB666, RGB565, YUV422_8 and YUV422_10 datatypes are also available for simulation and supported by testbench. RTL supports other data types as well.

2.3. Top-Level I/O

Table 2.3 shows the top-level I/O of this reference design. Actual I/O depends on the customer's channel and lane configurations. Compiler directives automatically declare all necessary I/O ports.

Table 2.3. CSI-2/DSI Duplicator Top-Level I/O

Port Name	Direction	Description
Clocks and Resets		
ext_clk_i (optional)	I	External input reference clock. Used to feed a clock to GPL and TX D-PHY PLL. Recommended to be minimum. 24 MHz. This port is declared only when RX_CLK_MODE_HS_LP is defined in synthesis_directives.v.
reset_n_i	I	Asynchronous active low system reset
CSI-2/DSI RX Interface		
rx_clk_p_i	I	Positive differential RX D-PHY input clock
rx_clk_n_i	I	Negative differential RX D-PHY input clock
rx_d0_p_i	I	Positive differential RX D-PHY input data 0
rx_d0_n_i	I	Negative differential RX D-PHY input data 0
rx_d1_p_i	I	Positive differential RX D-PHY input data 1 (in case of 2-lane or 4-lane configuration)
rx_d1_n_i	I	Negative differential RX D-PHY input data 1 (in case of 2-lane or 4-lane configuration)
rx_d2_p_i	I	Positive differential RX D-PHY input data 2 (in case of 4-lane configuration)
rx_d2_n_i	I	Negative differential RX D-PHY input data 2 (in case of 4-lane configuration)
rx_d3_p_i	I	Positive differential RX D-PHY input data 3 (in case of 4-lane configuration)
rx_d3_n_i	I	Negative differential RX D-PHY input data 3 (in case of 4-lane configuration)
CSI-2/DSI TX Interface		
tx0_clk_p_o	O	Positive differential TX D-PHY output clock on TX channel 0
tx0_clk_n_o	O	Negative differential TX D-PHY output clock on TX channel 0
tx0_d0_p_o	O	Positive differential TX D-PHY output data 0 on TX channel 0
tx0_d0_n_o	O	Negative differential TX D-PHY output data 0 on TX channel 0
tx0_d1_p_o	O	Positive differential TX D-PHY output data 1 on TX channel 0 (in case of 2/4-lane configuration)
tx0_d1_n_o	O	Negative differential TX D-PHY output data 1 on TX channel 0 (in case of 2/4-lane configuration)
tx0_d2_p_o	O	Positive differential TX D-PHY output data 2 on TX channel 0 (in case of 4-lane configuration)
tx0_d2_n_o	O	Negative differential TX D-PHY output data 2 on TX channel 0 (in case of 4-lane configuration)
tx0_d3_p_o	O	Positive differential TX D-PHY output data 3 on TX channel 0 (in case of 4-lane configuration)
tx0_d3_n_o	O	Negative differential TX D-PHY output data 3 on TX channel 0 (in case of 4-lane configuration)
tx1_clk_p_o	O	Positive differential TX D-PHY output clock on TX channel 1 (in case of 2 TX channel configuration)
tx1_clk_n_o	O	Negative differential TX D-PHY output clock on TX channel 1 (in case of 2 TX channel configuration)
tx1_d0_p_o	O	Positive differential TX D-PHY output data 0 on TX channel 1 (in case of 2 TX channel configuration)
tx1_d0_n_o	O	Negative differential TX D-PHY output data 0 on TX channel 1 (in case of 2 TX channel configuration)
tx1_d1_p_o	O	Positive differential TX D-PHY output data 1 on TX channel 1 (in case of 2 TX channel with 2/4-lane configuration)
tx1_d1_n_o	O	Negative differential TX D-PHY output data 1 on TX channel 1 (in case of 2 TX channel with 2/4-lane configuration)
tx1_d2_p_o	O	Positive differential TX D-PHY output data 2 on TX channel 1 (in case of 2 TX channel with 4-lane configuration)
tx1_d2_n_o	O	Negative differential TX D-PHY output data 2 on TX channel 1 (in case of 2 TX channel with 4-lane configuration)
tx1_d3_p_o	O	Positive differential TX D-PHY output data 3 on TX channel 1 (in case of 2 TX channel with 4-lane configuration)
tx1_d3_n_o	O	Negative differential TX D-PHY output data 3 on TX channel 1 (in case of 2 TX channel with 4-lane configuration)
tx2_clk_p_o	O	Positive differential TX D-PHY output clock on TX channel 2 (in case of 2 TX channel configuration)

Port Name	Direction	Description
tx2_clk_n_o	O	Negative differential TX D-PHY output clock on TX channel 2 (in case of 2 TX channel configuration)
tx2_d0_p_o	O	Positive differential TX D-PHY output data 0 on TX channel 2 (in case of 2 TX channel configuration)
tx2_d0_n_o	O	Negative differential TX D-PHY output data 0 on TX channel 2 (in case of 2 TX channel configuration)
tx2_d1_p_o	O	Positive differential TX D-PHY output data 1 on TX channel 2 (in case of 2 TX channel with 2/4-lane configuration)
tx2_d1_n_o	O	Negative differential TX D-PHY output data 1 on TX channel 2 (in case of 2 TX channel with 2/4-lane configuration)
tx2_d2_p_o	O	Positive differential TX D-PHY output data 2 on TX channel 2 (in case of 2 TX channel with 4-lane configuration)
tx2_d2_n_o	O	Negative differential TX D-PHY output data 2 on TX channel 2 (in case of 2 TX channel with 4-lane configuration)
tx2_d3_p_o	O	Positive differential TX D-PHY output data 3 on TX channel 2 (in case of 2 TX channel with 4-lane configuration)
tx2_d3_n_o	O	Negative differential TX D-PHY output data 3 on TX channel 2 (in case of 2 TX channel with 4-lane configuration)
tx3_clk_p_o	O	Positive differential TX D-PHY output clock on TX channel 3 (in case of 2 TX channel configuration)
tx3_clk_n_o	O	Negative differential TX D-PHY output clock on TX channel 3 (in case of 2 TX channel configuration)
tx3_d0_p_o	O	Positive differential TX D-PHY output data 0 on TX channel 3 (in case of 2 TX channel configuration)
tx3_d0_n_o	O	Negative differential TX D-PHY output data 0 on TX channel 3 (in case of 2 TX channel configuration)
tx3_d1_p_o	O	Positive differential TX D-PHY output data 1 on TX channel 3 (in case of 2 TX channel with 2/4-lane configuration)
tx3_d1_n_o	O	Negative differential TX D-PHY output data 1 on TX channel 3 (in case of 2 TX channel with 2/4-lane configuration)
tx3_d2_p_o	O	Positive differential TX D-PHY output data 2 on TX channel 3 (in case of 2 TX channel with 4-lane configuration)
tx3_d2_n_o	O	Negative differential TX D-PHY output data 2 on TX channel 3 (in case of 2 TX channel with 4-lane configuration)
tx3_d3_p_o	O	Positive differential TX D-PHY output data 3 on TX channel 3 (in case of 2 TX channel with 4-lane configuration)
tx3_d3_n_o	O	Negative differential TX D-PHY output data 3 on TX channel 3 (in case of 2 TX channel with 4-lane configuration)

3. Design and Module Description

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

- rx_dphy
- mipi_parser
- rx_buffer
- tx_ctrl
- tx_dphy_if
- tx_dphy

The top-level design has reset synchronization logic. In addition, GPLL may be added if necessary according to RX and TX configurations.

3.1. rx_dphy

This module must be created for RX channel according to channel conditions, such as the number of lanes, bandwidth, and others. Figure 3.1 shows an example of IP interface settings in the Lattice Radiant software for the CSI-2/DSI D-PHY Receiver Sub module IP version 1.9.0. Refer to CSI-2/DSI D-PHY Rx IP User Guide (FPGA-IPUG-02081) for details.

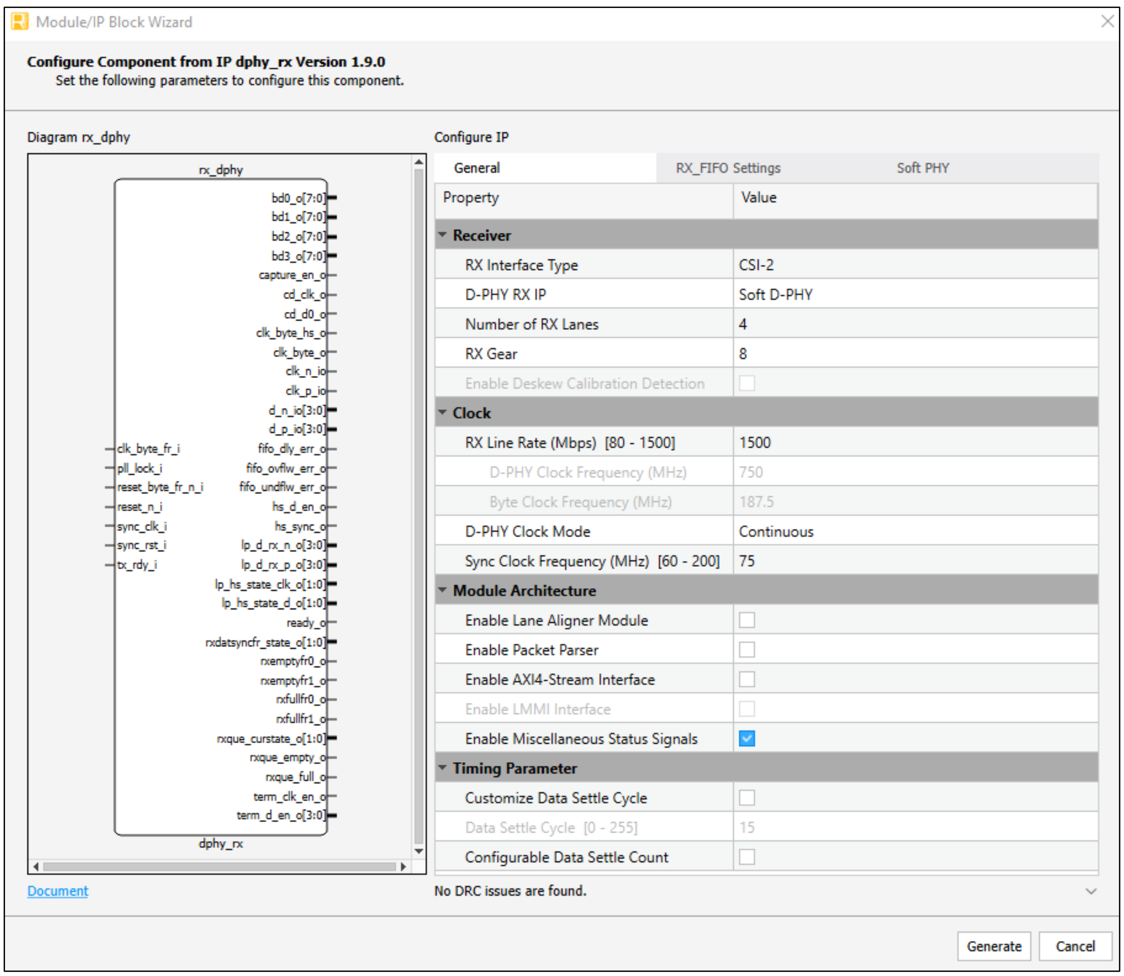

Figure 3.1. rx_dphy IP Creation in the Lattice Radiant Software

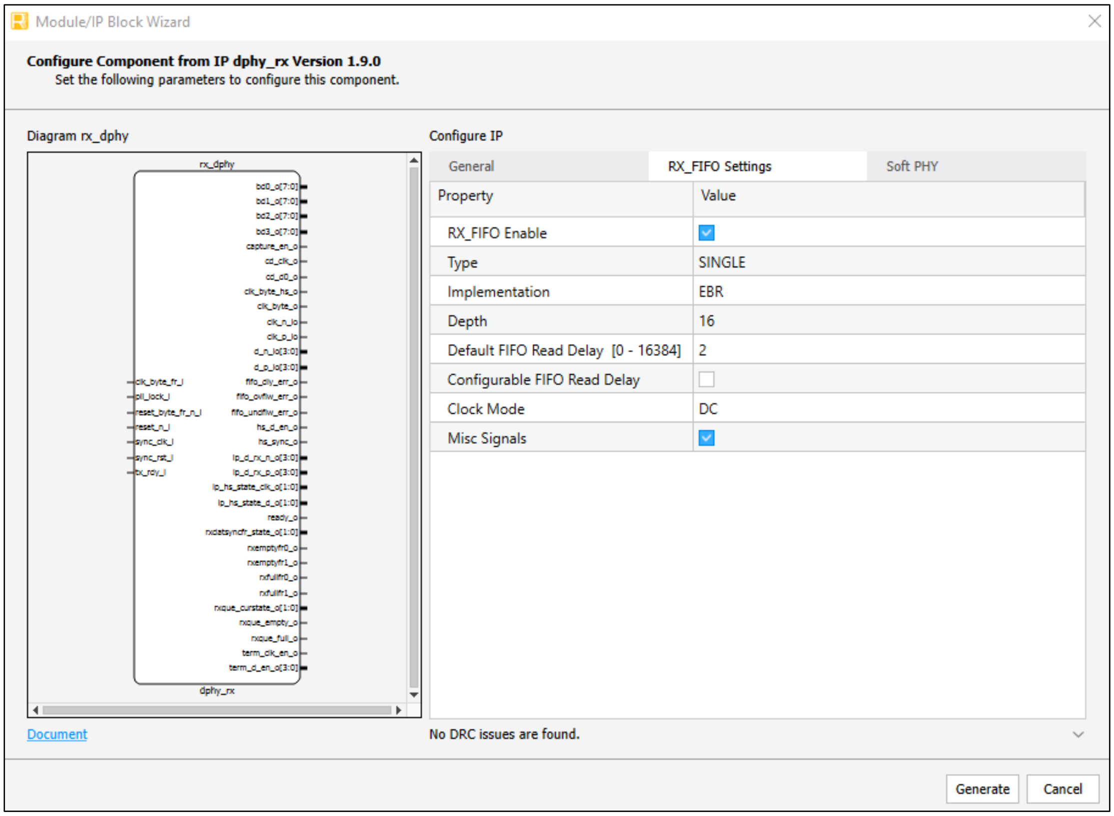

Figure 3.2. rx_dphy IP Creation Continued in the Lattice Radiant Software

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

- RX Interface – Select CSI-2 or DSI.
- DPHY RX IP – Select Hard D-PHY or Soft D-PHY according to channel configuration.
- Number of RX Lanes – Set according to channel configuration. The value must match NUM_RX_LANE_* setting.
- RX Gear – Select 8 or 16. 16 value is supported for 1-lane and 2-lane configurations only. 16 value is recommended when the RX byte clock speed exceeds 100 MHz with Gear 8 and STA fails.
- CIL Bypass – Always enable (checked) for RX HARDENED DPHY. For RX SOFT DPHY it is enabled by default.
- AXI Stream interface – Always disable (unchecked) for both RX HARDENED and SOFT DPHY channels.
- De-skew calibration detection – Disable (unchecked) for both RX HARDENED and SOFT DPHY channels when RX Line Rate is ≤ 1500 Mbps. It is automatically selected when the RX Line Rate is > 1500 Mbps.
- RX Line Rate – Set according to channel configuration.
- DPHY Clock Mode – Select Continuous or Non-continuous. Must match RX_CLK_MODE_* setting (Continuous = HS_ONLY, Non-continuous = HS_LP).
- Data Settle cycle – Recommended to use the suggested value. Try reducing this value one by one when RX D-PHY is not working as expected.
- Packet Parser – Always disable (unchecked) for both RX HARDENED and SOFT DPHY channels
- Miscellaneous Status signals – Always enable (checked) for both RX HARDENED and SOFT DPHY channels.
- Lane Aligner Module – Always enable (checked) for RX SOFT DPHY having two-lane and four-lane configurations. This option is not available for RX SOFT DPHY with one lane configuration.
- RX FIFO – Set according to channel configuration. For RX HARDENED DPHY it is enabled by default. For RX SOFT DPHY it can be enabled or disabled.
- RX FIFO Memory implementation - Set according to channel configuration.
- RX FIFO Depth - Set according to channel configuration. Minimum value 16 is required.
- RX FIFO Type - Set according to channel configuration.

- RX FIFO Packet delay - Set according to channel configuration. Minimum value 1 is required.
- RX Clock mode - Always set to DC (Dual clock) for RX SOFT DPHY. For RX HARDENED DPHY by default it is DC mode.
- RX FIFO Misc Signals- Always enable (checked) FIFO Miscellaneous signals for RX HARDENED and SOFT DPHY channels.
- In the Soft PHY tab, use the default settings with Edge Clock Centered Delay Mode.
- After configuring all required parameters, select Generate shown in the bottom right corner of the user interface.

3.1.1. RX FIFO

RX FIFO is useful especially in non-continuous clock mode and the continuous byte clock cannot have the exactly same frequency as the non-continuous byte clock used in D-PHY RX IP. It resides after the word aligner in case of Hardened D-PHY RX IP and resides before the word aligner in case of Soft D-PHY RX IP.

3.1.1.1. Hardened D-PHY in Continuous Clock Mode

Irrespective of CSI2 or DSI RX interface, in this case the minimum configuration of RX FIFO is recommended as mentioned, (LUT Memory based, Depth = 16, *Type* of Implementation = SINGLE, Packet Delay = 1, *Clock* Implementation = DC).

3.1.1.2. Soft D-PHY in Continuous Clock Mode

In this case, RX FIFO is not necessary and can be set to OFF.

3.1.1.3. Non-Continuous Clock Mode

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

- Continuous byte clock = non-continuous byte clock

  In this case, the minimum configuration of RX FIFO is recommended (LUT based, Depth = 16, *Type* Implementation = SINGLE, Packet Delay = 1, *Clock* Implementation = DC).
- Continuous byte clock < non-continuous byte clock

  In this case, *Type* Implementation = SINGLE and Packet Delay = 1 is recommended and others depend on the frequency ratio between these two clocks. When the clock speed difference is larger, the required depth of RX FIFO becomes larger. First, it is important to know the horizontal blanking period of the incoming RX channel. For example, in case that one-line active video period is 40 µs and the horizontal blanking is 4 µs, then we have 10 % of extra time to process the active data. This means the continuous byte clock can be as slow as ~-10% comparing to the non-continuous byte clock to avoid RX FIFO overflow.
- Continuous byte clock > non – continuous byte clock

  There are two options in this case:
  - Use *Type* Implementation = SINGLE with large Packet Delay

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

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

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

- Frequency relationship is unknown when the continuous byte clock is within the certain range against the non-continuous byte clock (for example, two clocks come from different clock sources which have ppm tolerance), we have no idea which clock is faster. The simplest way is to use *Type* Implementation = SINGLE with setting Packet Delay to the midpoint of FIFO depth when the tolerance is in ppm level. For example, assuming the clock tolerance is within +/- 500 ppm for 2 lane Gear 8 RAW10 with 1920 horizontal pixels. The payload byte count is 1920 x 5/4 = 2400 so that each lane takes 1200 bytes. 1200 x 500 ppm = 0.6, which means small FIFO with middle point read delay (ex. FIFO depth = 16 with read delay of 8-byte clock cycles) works fine. Using LUT is preferable as long as FIFO depth is small since using EBR here might cause the shortage of EBR in line_buf modules. QUEUE can also be used as described above even though it requires more EBRs.

  In case there is no detailed information regarding RX data (whether containing line start/end short packet, interval of the horizontal blanking period against active line period), the safest way is to set the continuous byte clock faster than the non-continuous byte clock. Use *Type* Implementation = QUEUE with Number of entries = 4 even though it might require more EBR resources comparing to *Type* Implementation = SINGLE, but ensure that total number of EBR used in the device do not exceed 20, the maximum number of EBRs available in CrossLink-NX.

3.2. mipi_parser

This module handles CSI-2/DSI protocol decoding. It detects the sync word (0xb8) and trailer bytes (0xff) to properly assert rx_bd_en, which indicates active packet data. Figure 3.3 shows an example of handling a CSI-2 Frame Start short packet. Data bytes *b8* and *ff* are trimmed from the incoming data by the rx_bd_en assertion (Num_RX_LANE_4, RX_GEAR_8). rx_bd_end indicates the end of the valid packet data.

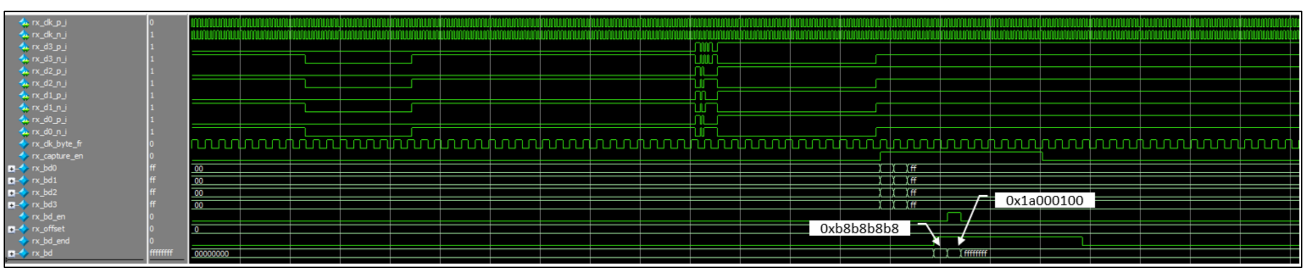

Figure 3.3. Active Data Detection (CSI-2 Frame Start)

Figure 3.4 shows the end of CSI-2 long packet detection. rx_bd_end is asserted in the last cycle of rx_bd_en. In this case, rx_offset = 0, which means LSB data (0x1a76) is not considered as active in the last rx_bd data indicated by rx_bd_en = 1 and rx_bd_end = 1. MSB 1 byte (0xff) is a trailer byte.

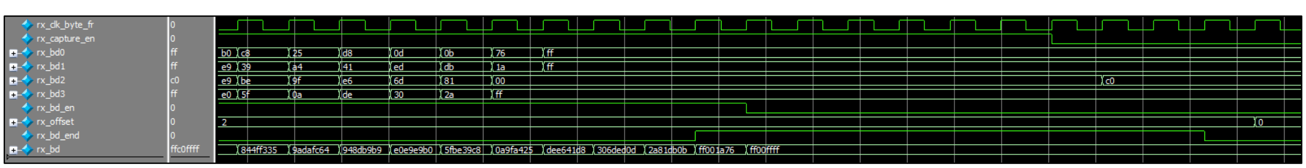

Figure 3.4. End of Active Data Detection (CSI-2 Long Packet)

3.3. rx_buffer

This module contains a dual clock FIFO using EBR to store valid packet data. Buffer depth is 512, 1024, 2048, or 4096 and data width is either 64 (when NUM_RX_LANE = 4 and RX Gear = 16) or 32 (others). This module itself with ready flag assertion reads out the first FIFO data. The FIFO has extra data width (other than 32 or 64 width as per the NUM_RX_LANE) to contain data end flag and offset data. Figure 3.5 shows the example of short packet write and read (NUM_RX_LANE = 4, FIFO data width is 32). rx_bd_en makes the FIFO write after the data width conversion from 16 bits to 32 bits and it also triggers the internal counter driven by rx_clk_byte_fr. When this counter reaches the value specified by BD_RD_DLY, the internal flag is asserted and that status is transferred to tx0_clk_byte domain, which makes buf_rdy = 1 to notify tx_ctrl module that the data are ready to be read. On the other hand, the 1st FIFO data is automatically read out by this module itself so that the 1st data is already available on buf_d when buf_rdy is asserted. In case of a short packet, the amount of active data is only 4 bytes and buf_d_end is also asserted along with short packet data of 0x1a000100 on buf_d. In this example BD_RD_DLY = 10 and buf_rdy is asserted soon after rx_bd_en = 1, but the data read (buf_re = 1) from the next module (tx_ctrl) does not begin due to LP to HS transition time on TX channel. This delay is expected to ~1 µs – 1.5 µs in case that both clock and data lanes requires LP to HS transition.

In general, the small value of BD_RD_DLY is preferable to make a process delay shorter, but it makes sense to set a larger value when there is some frequency difference or tolerance between rx_clk_byte_fr and tx0_clk_byte. If the clock ratio between these two is not exactly same as shown in Table 1.6, this FIFO could be used to absorb the frequency difference. Concept-wise, it is similar to SINGLE mode of RX FIFO in rx_dphy. When the tx0_byte_clk is faster than rx_byte_clk, setting BD_RD_DLY to a higher value (close to the rx_byte_clk cycle count to complete the payload data write for long packet in active video lines) helps to avoid FIFO underflow as long as the clock tolerance is absorbed during the LP period. On the other hand, setting a small value helps to avoid FIFO overflow when tx0_clk_byte is slower than rx_clk_byte_fr. If you are not clear regarding which clock is faster but tolerance is expected, then setting close to the middle value makes sense. The minimum buffer depth is 512 and that requires 2 EBRs (for data width 32) or 4 EBRs (for data width 64). This depth is expandable to 1024, 2048, or 4096 as long as the total number of EBR does not exceed 20.

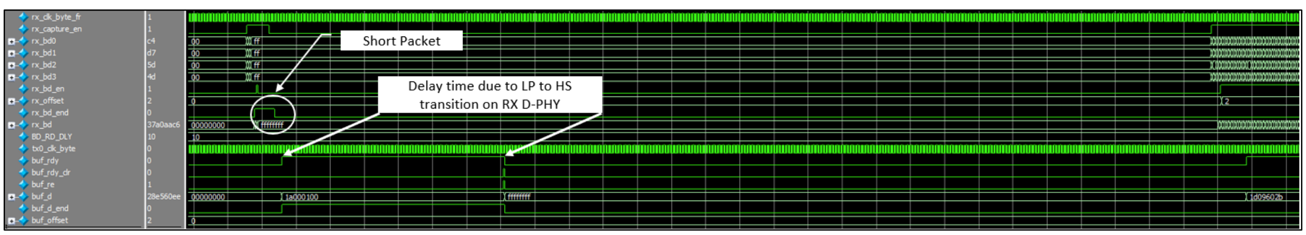

Figure 3.5. Short Packet Write and Read

Figure 3.6 and Figure 3.7 show the difference by BD_RD_DLY in the same design. In this case, WC (payload byte count) is 2400, which means it takes 1200 rx_clk_byte_fr cycles to write all long packet payload data to FIFO. tx0_clk_byte exactly matches the derived clock ratio by Table 1.6. In case of BD_RD_DLY = 800, FIFO read (buf_re = 1) begins while FIFO has some room to be filled up. In case of BD_RD_DLY = 1000, data is corrupted due to FIFO overflown shown by a spike of rx_fifo_full in Figure 3.7 inside a red box.

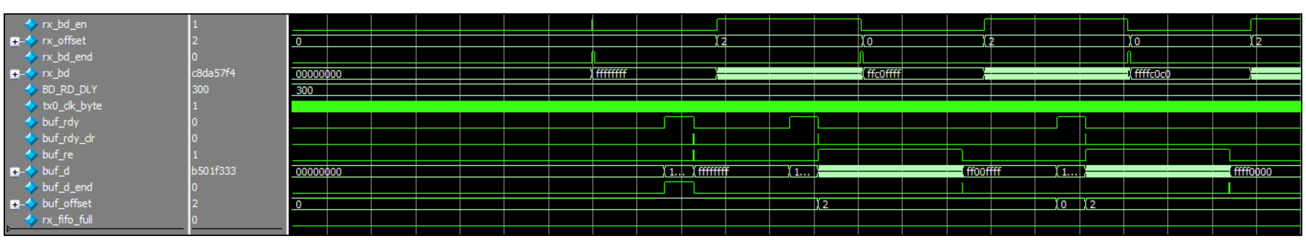

Figure 3.6. rx_buffer Write/Read Example with Depth = 512, BD_RD_DLY = 300

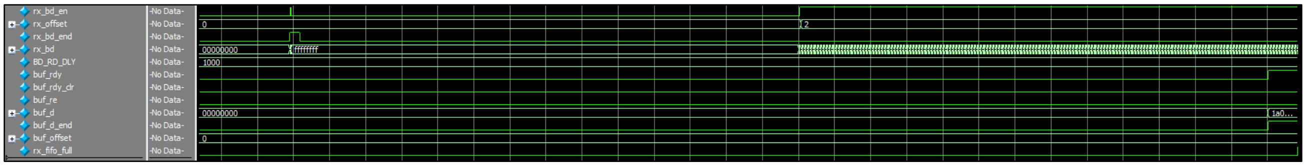

Figure 3.7. rx_buffer FIFO Overflow with Depth = 512, BD_RD_DLY = 1000

The safest way to avoid FIFO overflow in case that total TX bandwidth > RX total bandwidth (byte clock ratio value is larger than the ratio shown in Table 1.6) is to set the FIFO depth larger than the size of long packet data. Figure 3.8 shows the successful transaction setting the FIFO depth to 1024 with BD_RD_DLY = 1000. buf_rdy is asserted almost at the same timing of the completion of the one-line video data write to the FIFO. The FIFO depth of 1024 is deep enough to store the whole one-line data (1024 x 4 (bytes) > (4 (Header) + 2400 (payload) + 2 (CRC))) and FIFO overflow does not occur.

Figure 3.9 shows the example of successful transactions of (TX total bandwidth > RX total bandwidth). In this case, total TX bandwidth is twice of RX. The FIFO read period (buf_re = 1) is half of FIFO write period (rx_bd_en = 1), which leads more LP mode time on TX channel. This method (setting BD_RD_DLY close to the end of payload data write, setting FIFO depth to cover the long packet data size) works for any TX bandwidth as long as it is faster than RX bandwidth and does not exceed the maximum lane bandwidth of 1.5 Gbps in case of CSI-2, which mandates the data lane goes into LP mode after every packet transaction.

In case of DSI which keeps the data lane in HS mode during the horizontal blanking periods using null or blanking packets, this method cannot apply and TX bandwidth has to be exactly same (or close enough) to avoid FIFO overflow/underflow.

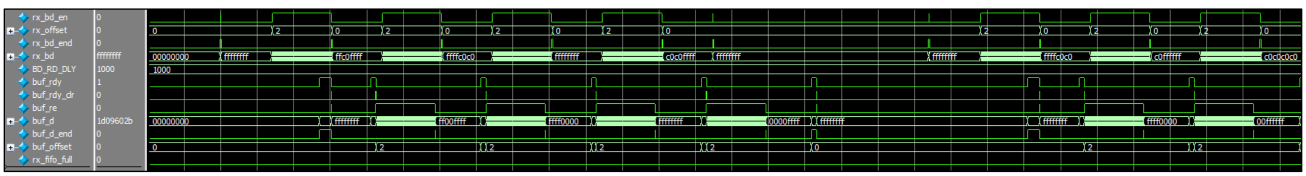

Figure 3.8. rx_buffer Successful Transactions with Depth = 1024, BD_RD_DLY = 1000

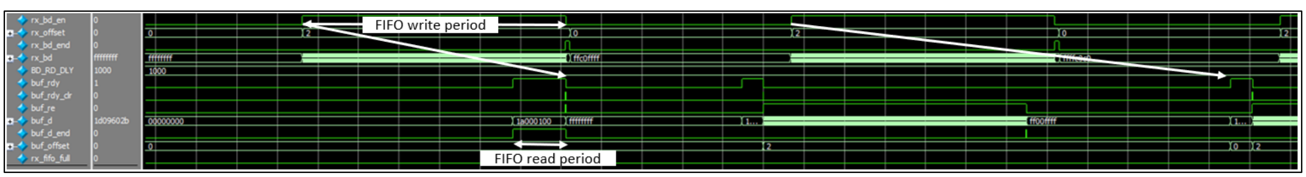

Figure 3.9. TX Bandwidth = 2 x RX Bandwidth with Depth = 1024, BD_RD_DLY = 1000

3.4. tx_ctrl

This module monitors the read ready flag of rx_buffer and then reads RX Buffer data. The read data are sent to tx_dphy through tx_dphy_if. Figure 3.10 shows an example of a short packet transaction. After receiving tx_bgn from tx_dphy_if, this module begins capturing buf_d and sends these on tx_bd with proceeding *B8* (sync word) and appending *00* or *FF* (trailer byte).

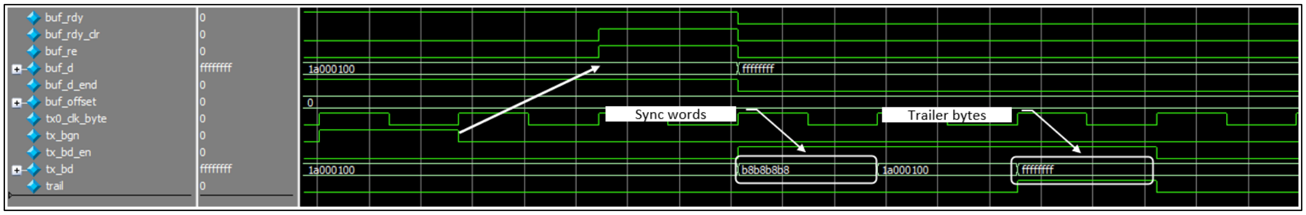

Figure 3.10. Short Packet Transaction (NUM_TX_LANE_4, TX_GEAR_8)

Figure 3.11 shows an example of the end of long packet transaction. In this case, the lane configuration between RX and TX is different so that trailer bytes are recreated and appended by this module.

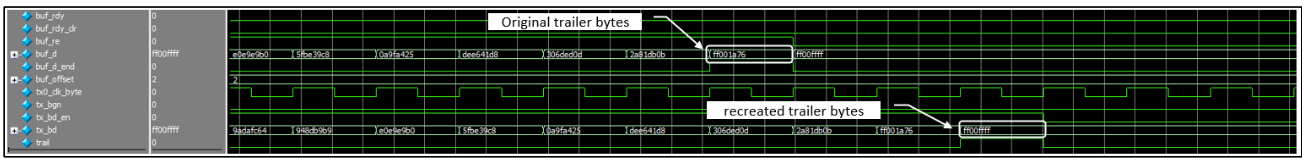

Figure 3.11. End of Long Packet Transaction (NUM_RX_LANE_4, RX_GEAR_8 to NUM_TX_LANE_2, TX_GEAR_8)

When TX D-PHY is set to non-continuous clock mode, it is possible to keep the clock lane in HS mode during the horizontal blanking periods. This is enabled when KEEP_HS is defined in synthesis_directives.v. This feature is useful when the horizontal blanking period is not long enough to have both clock and data lanes go into LP mode, which requires more overhead time. In that case, this module keeps clk_hs_en = 1 during the active video period including the horizontal blanking periods and makes clk_hs_en = 0 only during the vertical blanking period so that the clock lane goes into LP mode only during the vertical blanking period. This feature is available only for CSI-2.

3.5. tx_dphy_if

This module resides between tx_ctrl and tx_dphy to transfer control signals and data including data bus allocation and clock domain conversion in case of two or more channels TX outputs. Figure 3.12 shows an example of short packet transaction in non-continuous clock mode (only data lane 0 is shown in the figure).

1. Wait for buf_rdy = 1
2. Check tx0_c2d_rdy = 1, then assert tx0_clk_hs_en and tx0_d_hs_en (at least one tx0_byte_clk cycle)
3. Clock lane goes into HS mode
4. Data lane goes into HS mode
5. Wait for tx0_d_hs_rdy = 1
6. Assert tx_bgn
7. FIFO read happens by tx_ctrl
8. Receive HS data by tx_bd_en
9. Send HS data to tx_dphy along with tx0_dphy_pkten = 1
10. HS data transmission by tx_dphy
11. After HS transmission is done, tx0_d_hs_rdy goes 0 and data lane goes into LP mode
12. Clock lane goes into LP mode

13. After all HS transaction ends, tx0_c2d_rdy becomes 1, which means tx_dphy is ready to handle the next HS transaction.

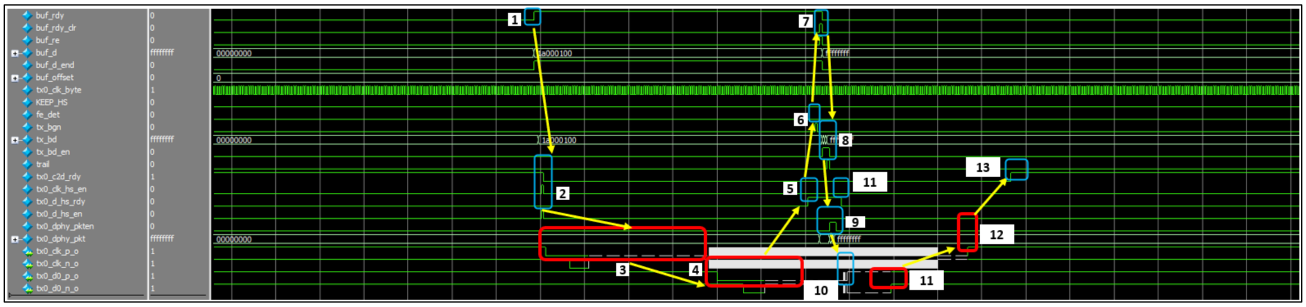

Figure 3.12. Short Packet Transaction in Non-Continuous Clock Mode

In case of CSI-2 in non-continuous clock mode, it is possible to keep clock lane in HS mode during the horizontal blanking periods by defining KEEP_HS in synthesis_directives.v. Figure 3.13 shows the global sequence without defining KEEP_HS and Figure 3.14 shows the case with KEEP_HS. When KEEP_HS is defined, the control logic for tx0_clk_hs_en takes fe_det (Frame End detect) and keeps tx0_clk_hs_en = 1 during the horizontal blanking periods.

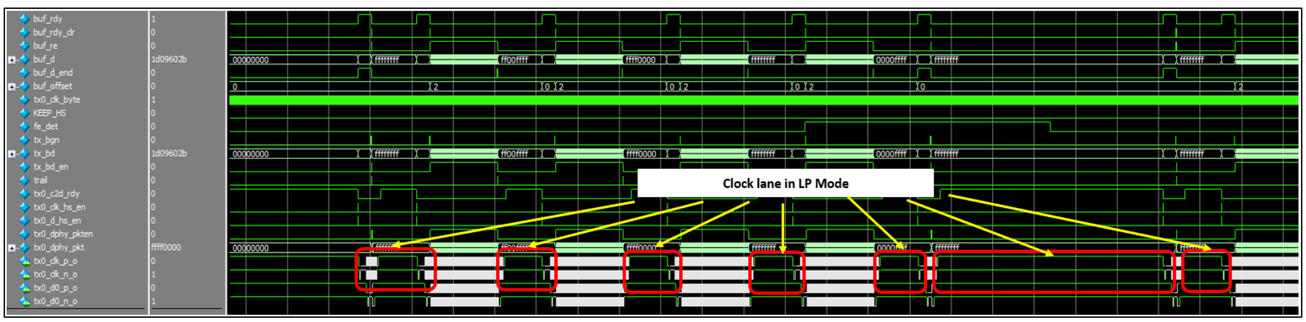

Figure 3.13. Global Sequence in Non-Continuous Clock Mode without KEEP_HS

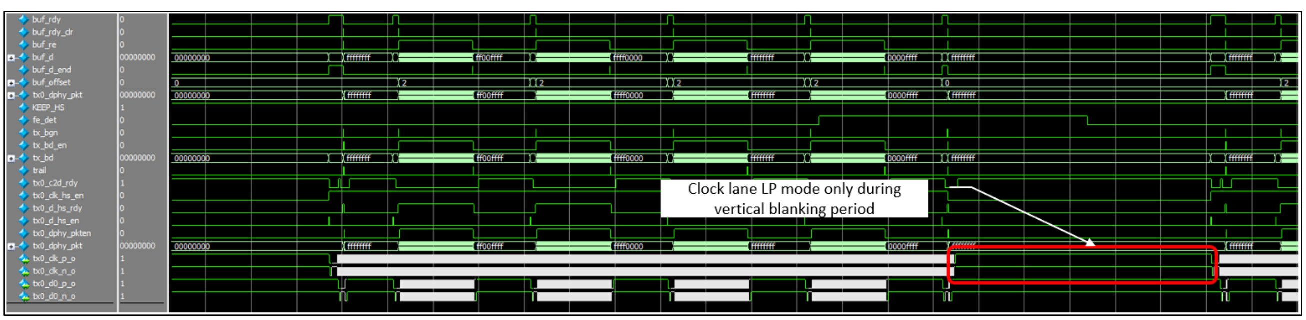

Figure 3.14. Global Sequence in Non-Continuous Clock Mode with KEEP_HS

This module takes care of four TX channels by communicating with four tx_dphy modules. Operations are almost similar to one TX channel output except for clock domain conversions, which lead to a few cycle delays in transactions.

3.6. tx_dphy

You must create this module according to channel conditions, such as number of lanes, bandwidth, and others. Figure 3.15 shows an example of IP interface setting in the Lattice Radiant software for the CSI-2/DSI D-PHY Transmitter Submodule IP. Refer to CSI-2/DSI D-PHY Tx IP User Guide (FPGA-IPUG-02080) for details.

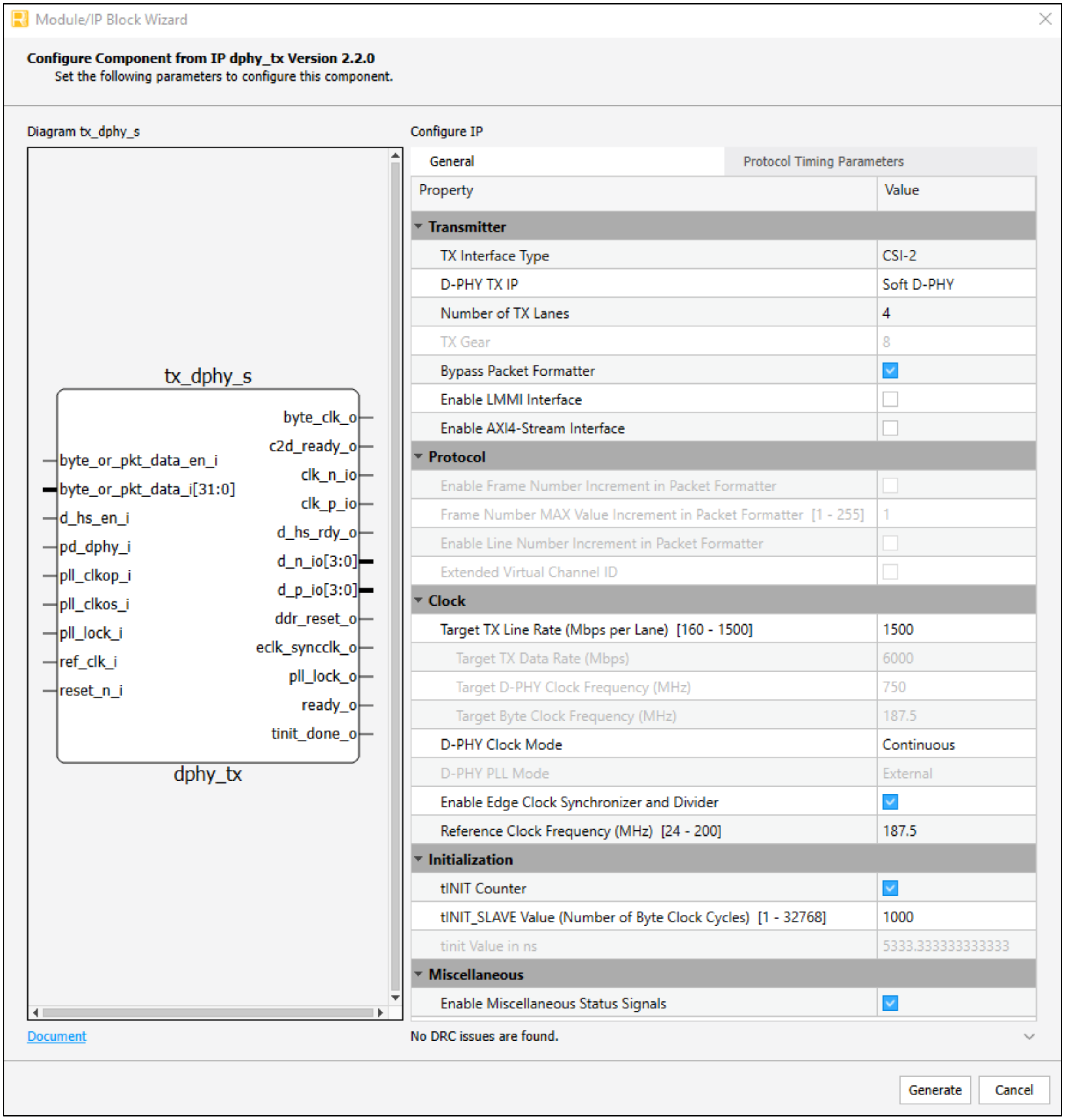

Figure 3.15. tx_dphy IP General Parameters Creation in the Lattice Radiant Software

The following are the guidelines for General parameter settings required for this reference design.

- TX Interface Type – Select CSI-2 or DSI.
- DPHY TX IP – Set as per channel configuration.
- Number of TX Lanes – Set according to channel configuration. Must match NUM_TX_LANE_* setting.
- TX Gear – Set according to TX Line Rate. In general, 16 must be selected when TX Line Rate is 900 or higher.
- CIL Bypass – Always enable (checked) for TX HARDENED DPHY. For TX SOFT DPHY, it is enabled by default.

- Bypass packet formatter – Always enable (checked) for TX HARDENED and SOFT DPHY channels.
- LMMI Interface – Always disable (unchecked) for TX HARDENED and SOFT DPHY channels
- AXI Stream Interface – Always disable (unchecked) for TX HARDENED and SOFT DPHY channels
- Periodic Skew calibration – Always disable (unchecked) for TX HARDENED and SOFT DPHY channels
- Target TX Line Rate – Set according to channel configuration. This value must be equal to or larger than (number of RX channel) x (RX channel bandwidth) / (number of TX lanes).
- D-PHY Clock Mode – Set according to channel configuration. Select Continuous when the horizontal blanking period is short in DSI. In case of CSI-2, Non-continuous and defining KEEP_HS could be an option for short horizontal blanking.
- DPHY PLL Mode - Set according to channel configuration.
- Reference Clock Frequency – Set the appropriate value, which can be obtained from ext_clk_i pin, continuous rx_byte_clk_fr, or on-chip GPLL. This clock frequency must be in the range of 24-200 MHz.
- tINIT Counter – Always enabled (checked) for TX HARDENED and SOFT DPHY channels.
- tINIT_SLAVE Value – Set to 1000 (default) is recommended.
- Miscellaneous status signals – Always enabled (checked) for TX HARDENED and SOFT DPHY channels.
- After configuring required General parameters, select Generate shown in the bottom right corner of user interface.

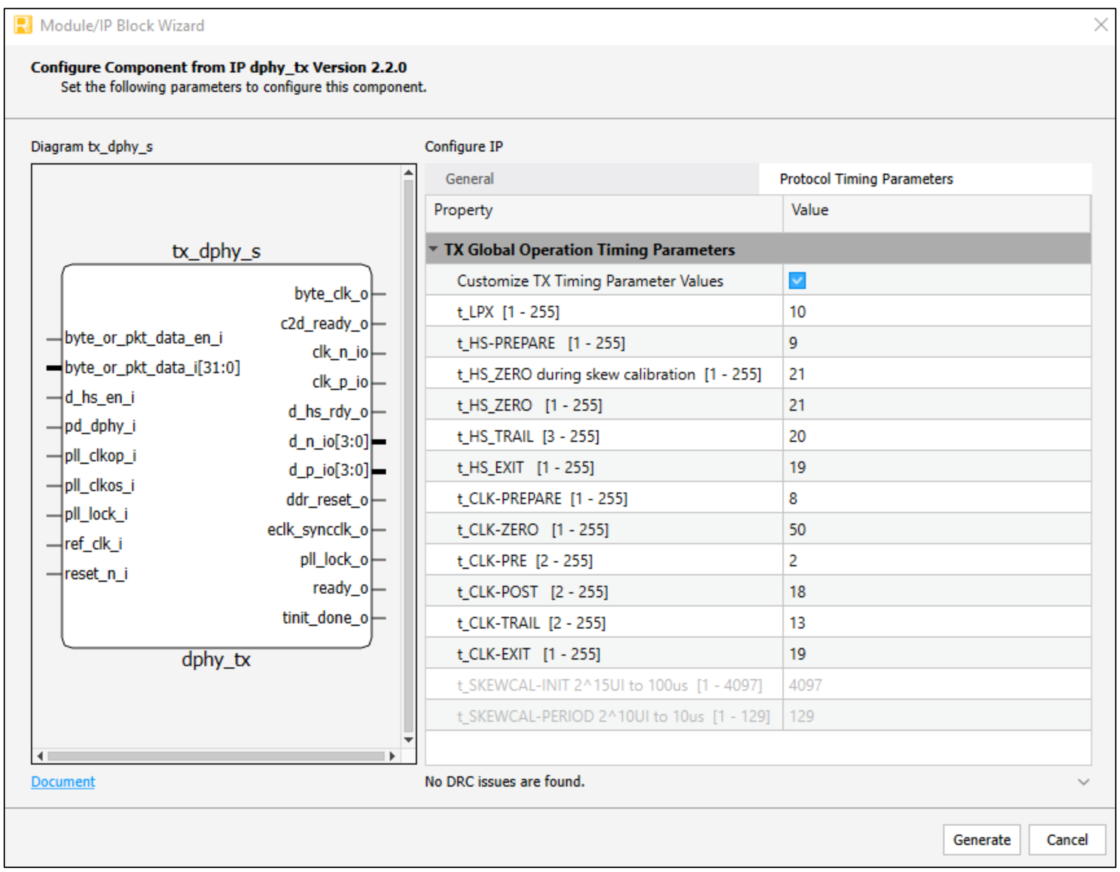

Figure 3.16. tx_dphy IP Protocol Timing Parameters Creation in the Lattice Radiant Software

The following are the guidelines for Protocol timing parameter settings required for this reference design.

- While generating a TX DPHY IP in the Lattice Radiant software, the Protocol timing parameters are set to appropriate values according to the TX DPHY General Clock settings as shown in Figure 3.16.
- If required to customize these values while creating TX DPHY IP, after configuring all General parameters, go to the Protocol Timing Parameters window, and enable (check) option Customize TX Timing Parameter Value. After changing the desired parameter value, Select Generate in user interface shown at the bottom left corner.

- For any configuration if DPHY LP00+HS00 transition Period is less than 300ns, check t_CLK_ZERO value and modify such that (t_CLK_ZERO value * (TX Byte clock period in ns)) >= 300 ns. After changing the value, select Generate in user interface shown at the bottom right corner.
- For more information refer to CSI-2/DSI D-PHY Tx IP User Guide (FPGA-IPUG-02080).

TX DPHY module takes the byte data and outputs DSI/CSI-2 data after serialization in HS mode. It is recommended to set the design name to *tx* and module name to *tx_dphy_h for HARDENED DPHY or tx_dphy_s for SOFT DPHY* so that you do not need to modify the instance name of this IP in the top-level design as well as the simulation setup file. Otherwise, you need to modify the names accordingly.

General guideline of TX Gear setting is to set 8 when the lane bandwidth is less than 900 Mbps, which means TX byte clock could be ~112 MHz. If this causes timing violations in Static Timing Analysis (STA), TX Gear should be changed to 16.

Be aware of the relationship between the reference clock and DPHY clock. DPHY clock is generated by the internal PLL of TX D-PHY IP. Following is the equation to generate DPHY clock:

$$TX\_Line\_Rate\_per\_lane = \frac{1}{NI} * ref\_clk\_frequency * \frac{M}{NO}$$

where NI = 1, 2, 3, 4, or 5; M = 16, 17, …, or 255; NO = 1, 2, 4, or 8. The following restrictions also exist:

$$24MHz \leq \frac{1}{NI} * ref\_clk\_frequency \ \leq 30\ MHz,$$

$$640MHz \leq \frac{1}{NI} * ref\_clk\_frequency * M \leq 1500MHz$$

Set the appropriate TX Line Rate (per lane) which can be obtained by the above equations applying the given reference clock frequency.

3.7. Clock Distribution

In case of non-continuous clock (HS_LP) mode, two clock trees (non-continuous RX byte clock and continuous RX byte clock) exist in CrossLink-NX device. The following are possible candidates of the continuous clock in case of non-continuous clock mode:

- PLL outputs driven by the external reference clock
- Continuous clock by the external reference clock (either direct or after divider)

The sample design (mipi_duplicator_NX.v) has typical clock assignments to rx_clk_byte_fr and rx_clk_lp_ctrl according to the clock mode on RX channel. The code snippet is shown below. rx_clk_lp_ctrl (clock signal for LP and HS mode control module for clock lane) could be different from rx_clk_byte_fr (continuous byte clock for RX channel), but recommended to be the same to save the primary clock tree resources. *int_gpll* is the name used in this top-level design. This name has to be changed if the different name is used.

```
//////////////////////////////////////////////////////////////////////////////////
///// Clock distribution for Rx/Tx channels Byte clock
//////////////////////////////////////////////////////////////////////////////////

`ifdef RX_CLK_MODE_HS_ONLY
       assign rx_clk_byte_fr = rx_clk_byte_hs;
       assign rx_clk_lp_ctrl = 1'b1;
`else
// User must provide the continuous RX byte clock from ext_clk_i or somewhere
//     assign rx_clk_byte_fr = pll_clkop;
//     assign rx_clk_byte_fr = pll_clkos;
       assign rx_clk_byte_fr = pll_clkos2;
       assign rx_clk_lp_ctrl = ext_clk_i;
`endif

//////////////////////////////////////////////////////////////////////////////////
///// GPLL instantitation
//////////////////////////////////////////////////////////////////////////////////
`ifdef USE_GPLL
       int_gpll int_gpll (
              .rstn_i (1'b1),
       `ifdef RX_CLK_MODE_HS_LP
              .clki_i    (ext_clk_i),
       `else
              .clki_i    (rx_clk_byte_fr),
       `endif
              .clkop_o   (pll_clkop),      // Tx HARD PLL clock
       `ifdef RX_CLK_MODE_HS_LP
              .clkos_o   (pll_clkos),      // Tx SOFT PLL clock
              .clkos2_o  (pll_clkos2),     // Rx DPHY Byte clock
       `endif
 //           .clkos3_o  (pll_clkos3),     // Tx SOFT PLL clock 90-Deg
              .lock_o    (gpll_lock)
       );
`endif

//////////////////////////////////////////////////////////////////////////////////
```

On TX side, using continuous or non-continuous clock mode does not affect the number of necessary clock trees (always uses one clock tree per TX channel). In general, using the same clock mode as RX side is recommended. Special care is needed when configured to use non-continuous mode on TX side while RX side uses continuous mode since LP-HS transition time on RX side might not be long enough to allow TX side to makes LP-HS transitions, including the clock lane, which leads to FIFO overflow. To feed a clock to TX D-PHY IP, the external clock is necessary if continuous RX byte clock is not appropriate to generate the desired clock for TX D-PHY. The clock to TX D-PHY must be continuous and within the range 24-200 MHz.

4. Design and File Modifications

Depends on the clocking scheme, some modifications are required depending on your configuration in addition to two directive files (synthesis_directives.v, simulation_directives.v).

4.1. Top-Level RTL

The current top-level file (mipi_duplicator_NX.v) includes sample assignments and GPLL instantiation according to RX clock mode and external clock availability defined in synthesis_directives.v as described in Clock Distribution section. This part may need to be modified depending on the clocking scheme.

In addition, instance names of RX/TX D-PHY (rx_dphy, tx_dphy_h or tx_dphy_s) should be modified if you created these IP with different names.

5. Design Simulation

The script file (mipi_dup_NX_msim.do) and testbench files are provided to run the functional simulation using the QuestaSim Lattice Edition FPGA Edition software 2024.2. You have to launch the QuestaSim software from the Lattice Radiant software. If you follow the naming recommendations regarding design name and instance name when RX and TX D-PHY IPs are created by the Lattice Radiant software, the following are the only changes required in the script file.

- Lattice Radiant installation directory path
- Your project directory

```
### Set Radiant installation directory ###
set radiant_dir C:/lscc/radiant/3.1

### Set Customer's simulation directory ###
set work_dir
C:/Users/gquiriad/Documents/User_Guide/Duplicator_CrosslinkNX/RD/1_to_N_Duplicator_Crosslink_NX/mipi_dup_RD_NX/simulation/lifcl

cd $work_dir
```

Lattice Radiant directory

User project directory

Figure 5.1. Script Modification #1

```
if {![file exists work]} {
    vlib work
}
transcript file "simulation.log"

vmap work work

vmap ovi_lifcl $radiant_dir/modeltech/lib/ovi_lifcl
vmap pmi_work $radiant_dir/modeltech/lib/pmi_work

vlog -mfcu \
+incdir+./../../testbench/verilog+./../../source/verilog/lifcl
./../../testbench/verilog/mipi_duplicator_NX_tb.v \
-y ./../../testbench/verilog +libext.v \
./../../source/verilog/lifcl/mipi_duplicator_NX.v \
./../../source/verilog/lifcl/mipi_parser.v \
./../../source/verilog/lifcl/rx_buffer.v \
./../../source/verilog/lifcl/tx_ctrl.v \
./../../source/verilog/lifcl/tx_dphy_if.v \
./../../source/verilog/lifcl/clk_xfer.v \
./../../rx_dphy/rtl/rx_dphy.v \
./../../tx_dphy_h/rtl/tx_dphy_h.v \
./../../tx_dphy_s/rtl/tx_dphy_s.v \
./../../int_gpll/rtl/int_gpll.v \

##./../../int_osc/rtl/int_osc.v \

vsim -t 10fs -voptargs=+acc -L work top -L pmi_work -L ovi_lifcl
do wave.do
run -all
```

Figure 5.2. Script Modification #2

You need to modify simulation_directives.v according to your configuration (refer to Simulation Directives for details). By executing the script in the QuestaSim software, compilation and simulation are executed automatically. The testbench takes all data comparison between the expected data and output data from the RD, including VC, ECC and CRC data. It shows following statements while running and doing data comparison.

```
Check the Packet Header to identify incoming RX channel --- 2b 60 09 1d matches RX Data
#          19127306200 DPHY RX CH Driving Data
#          19127306200 DPHY RX CH Lane 0 : Driving with data = fc
#          19127306200 DPHY RX CH Lane 1 : Driving with data = 08
#          19127306200 DPHY RX CH Lane 2 : Driving with data = 53
#          19127306200 DPHY RX CH Lane 3 : Driving with data = 88
# [19127441250][DPHY_TX3_CHK] payload data = c3 f8 64 70 --- [19127441250][DPHY_TX3_CHK]Data
matches RX Data : c3 f8 64 70
# [19127441250][DPHY_TX2_CHK] payload data = c3 f8 64 70 --- [19127441250][DPHY_TX2_CHK]Data
matches RX Data : c3 f8 64 70
# [19127441250][DPHY_TX1_CHK] payload data = c3 f8 64 70 --- [19127441250][DPHY_TX1_CHK]Data
matches RX Data : c3 f8 64 70
# [19127643810][DPHY_TX0_CHK] Long Packet detected : Data type = 2b
# Check the Packet Header to identify incoming RX channel --- 2b 60 09 1d matches RX Data
#          19127846200 DPHY RX CH Driving Data
#          19127846200 DPHY RX CH Lane 0 : Driving with data = 59
#          19127846200 DPHY RX CH Lane 1 : Driving with data = 48
#          19127846200 DPHY RX CH Lane 2 : Driving with data = 84
#          19127846200 DPHY RX CH Lane 3 : Driving with data = 5b
```

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

```
# [19742467500][DPHY_TX3_HS_LP_CHK] HS-TRAIL period is too short or too long on D0 lane or
clock lane is not in HS mode!!! HS_TRAIL period = 0 ns, must be between 62 and 113 ns!!!
# [19758566310][DPHY_TX0_HS_LP_CHK] HS to LP11 Transition on clock lane with TRAIL period =
67 ns
# [19759747500][DPHY_TX1_HS_LP_CHK] HS to LP11 Transition on clock lane with TRAIL period =
70 ns
# [19759747500][DPHY_TX2_HS_LP_CHK] HS to LP11 Transition on clock lane with TRAIL period =
70 ns
# [19759747500][DPHY_TX3_HS_LP_CHK] HS to LP11 Transition on clock lane with TRAIL period =
70 ns
#           21868852500 DPHY RX CH CLK CONT : Driving CLK-Trail
#           21874817500 TEST END
#
# TX CH #0 : 18 / 18 HS transmission completed successfully
# TX CH #1 : 18 / 18 HS transmission completed successfully
# TX CH #2 : 18 / 18 HS transmission completed successfully
# TX CH #3 : 18 / 18 HS transmission completed successfully
```

One HS transmission is most likely either Frame Start/End short packet or long packet of one active video line in case of CSI-2. In case of DSI, one HS transmission might include multiple short and long packets. The result is if the numerator is equal to denominator in the statements.

You should set small values in NUM_LINES and NUM_FRAMES directives in simulation_directives.v file, especially in the first simulation trial to minimize simulation time. On the other hand, it makes sense to set the actual value to NUM_PIXELS directives and RX_DPHY_LPS_GAP directives when TX bandwidth cannot have extra margin against total RX bandwidth. By setting realistic values, FIFO overflow is detected when the margin is not large enough.

Figure 5.3 shows an example of CSI-2 duplication of four TX channels with the RX channel in continuous clock mode and TX channels in non-continuous clock mode. RX and TX both are 4-lane. RX and TX lane bandwidth is 1480 Mbps, which means total bandwidth of RX and TX channel is (1480x =4 lane=5920 Mbps). In this case, BD_RD_DLY is set to 10. As shown in the figure for TX0 channel, similar behavior can be also seen for remaining three Channels TX1, TX2, and TX3.

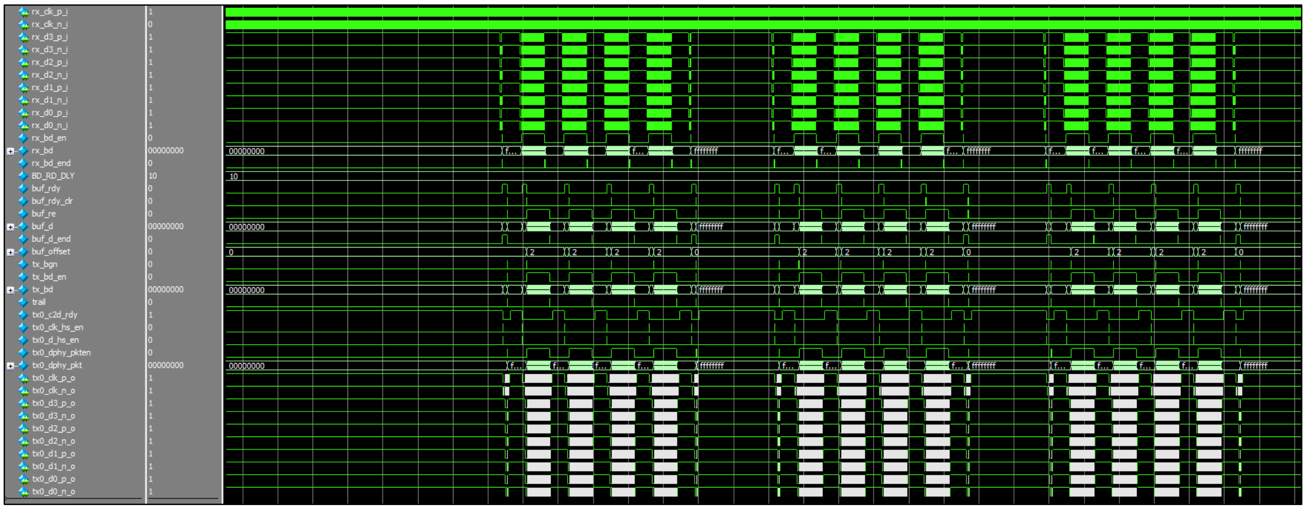

Figure 5.3. Functional Simulation Example of CSI-2 with Four Non-Continuous Clock Mode TX Channels

6. Known Limitations

The following are the limitations of this reference design:

- Data lane(s) must go into LP (Low Power) mode during horizontal blanking periods.
- In case of two or more channel output, all the TX channels must have the same configuration.

7. Design Package and Project Setup

1 to N MIPI CSI-2/DSI Duplicator Reference Design for CrossLink-NX is available on the Lattice Semiconductor web page. Figure 7.1 shows the directory structure. The design is targeted for LIFCL-40-7BG400C. The reference design is to duplicate single MIPI CSI2 RX Channel to four MIPI CSI2 TX Channels. synthesis_directives.v and simulation_directives.v are set to configure single-lane RX and four-lane TX channels of the design as shown below:

- RX CH – 4 lane with Hardened D-PHY with Gear 8 in continuous clock mode
- TX CH #0 – 4 lane with Hardened D-PHY with Gear 8 in Non-continuous clock mode using Internal PLL mode
- TX CH #1 – 4 lane with Soft D-PHY with Gear 8 in Non-continuous clock mode using External PLL mode
- TX CH #2 – 4 lane with Soft D-PHY with Gear 8 in Non-continuous clock mode using External PLL mode
- TX CH #3 – 4 lane with Soft D-PHY with Gear 8 in Non-continuous clock mode using External PLL mode

You can modify the directives for own configuration.

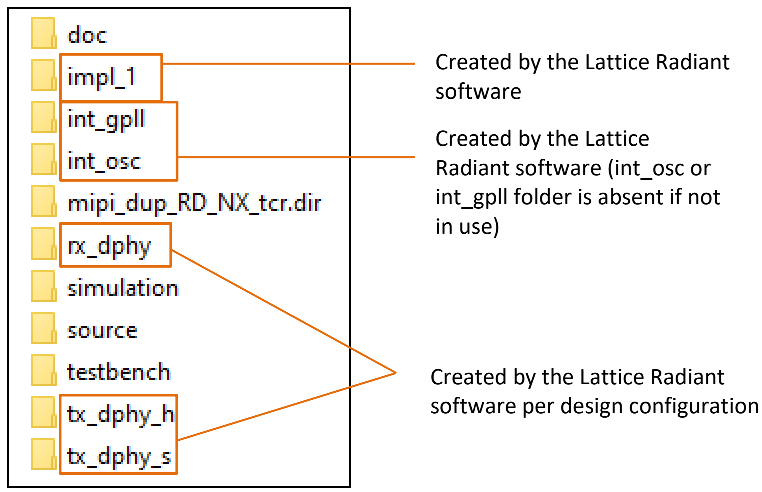

Figure 7.1. Directory Structure

The Lattice Radiant Design IP folders namely *rx_dphy*, *tx_dphy_h, tx_dphy_s, int_gpll and int_osc* are included under the project directory when created from IP Catalog of the Lattice Radiant software.

For RX DPHY, the closed-box files (*_bb.v) are contained in the rx_dphy/rtl/ folder. Similarly, for other IPs generated from the Lattice Radiant software IP Catalog, the closed-box files are present in their respective /rtl/ folder. Figure 7.2 shows design files used in this Lattice Radiant project. The Lattice Radiant software creates five (.ipx) files. By keeping the synthesis_directives.v file on top in the Input Files Folder and specifying mipi_duplicator_NX.v as a top-level design, all necessary files are included and they can be seen in the Design hierarchy window only if the design is loaded without any errors. Required (.ipx) files must be added or generated from the Lattice Radiant software IP Catalog if necessary.

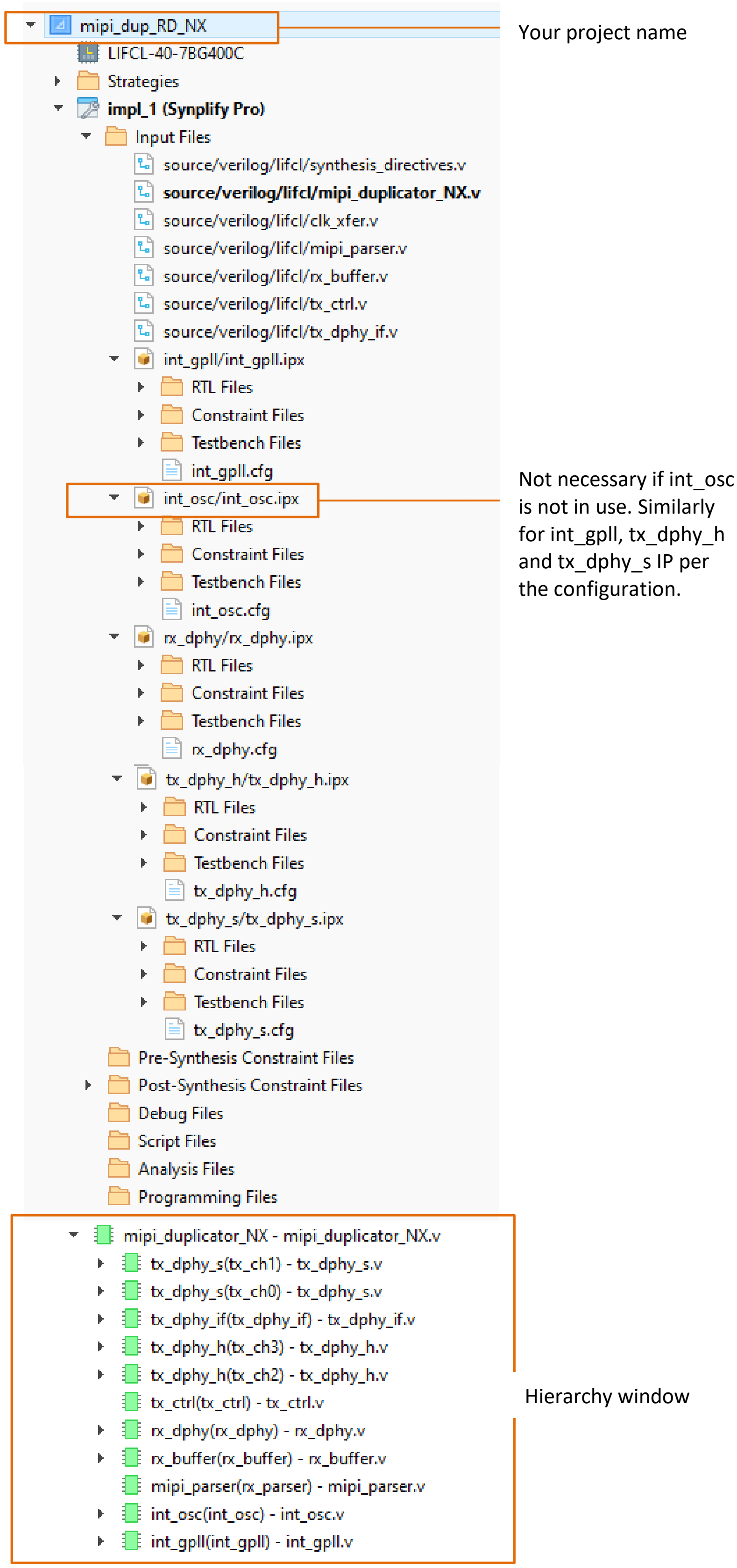

Figure 7.2. Project Files in the Lattice Radiant Software

8. Resource Utilization

Resource utilization depends on the configurations and IP version. Table 8.1 shows resource utilization of this Reference Design configuration. Actual usage may vary.

Table 8.1. Resource Utilization for the Reference Design

Configuration	LUT %	FF %	EBR	I/O
CSI2 RX channel with 1-lane, Gear 8 to Four CSI2 TX channels with 4-lanes, Gear 8	6	5	1	26
CSI2 RX channel with 1-lane, Gear 16 to Four CSI2 TX channels with 1-lane, Gear 8	5	3	1	14
DSI RX channel with 4-lanes, Gear 8 to Three DSI TX channels with 4-lanes, Gear 8	5	5	1	22
DSI RX channel with 1-lane, Gear 8 to Two DSI TX channels with 2-lanes, Gear 8	4	3	1	6

References

- MIPI Alliance Specification for D-PHY Version 1.2
- MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2) Version 1.2
- MIPI Alliance Specification for Display Serial Interface (DSI) Version 1.2
- CSI-2/DSI D-PHY Rx IP User Guide (FPGA-IPUG-02081)
- CSI-2/DSI D-PHY Tx IP User Guide (FPGA-IPUG-02080)
- CrossLink-NX web page
- 1 to N MIPI CSI-2/DSI Duplicator Reference Design web page
- Lattice Radiant Software web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.2, March 2025

Section	Change Summary
All	Performed minor editorial fixes.
Disclaimers	Updated disclaimers.
Inclusive Language	Added inclusive language boilerplate.
Introduction	• Updated the compatible IP versions supported in Table 1.1. Supported Device and IP. • Updated the description on clocking scheme examples for external PLL mode in the Block Diagram and Clocking Scheme section.

Design and Module Description	• Updated IP version and added description on the Soft PHY tab in the rx_dphy section.
	• Updated the following figures:
	  • Figure 3.1. rx_dphy IP Creation in the Lattice Radiant Software
	  • Figure 3.2. rx_dphy IP Creation Continued in the Lattice Radiant Software
	  • Figure 3.15. tx_dphy IP General Parameters Creation in the Lattice Radiant Software
	  • Figure 3.16. tx_dphy IP Protocol Timing Parameters Creation in the Lattice Radiant Software
	• Removed figure: *tx_dphy IP General Parameters Creation Continued in Lattice Radiant Software*.
	• Updated the code in the Clock Distribution section.

Design Simulation	Changed *ModelSim* to *QuestaSim* in the Design Simulation section.
Design Package and Project Setup	Changed *blackbox* to *closed-box* in the Design Package and Project Setup section.
Resource Utilization	Mentioned that the resource utilization depends on IP version.
References	Updated references.

Revision 1.1, March 2023

Section	Change Summary
Introduction	• Updated the D-PHY Receiver IP version 1.1.1 to 1.4.3 and D-PHY Transmitter IP version from 1.1.3 to 1.7.2 in Supported Device and IP section. • Added sentence *The excel file for the parameter calculator is located to the design package source code. To locate mipi_dup_RD_NX.xlsx, open the source code folder and locate the folder "docs".*
Design and Module Description	Updated below figures as per D-PHY Receiver IP version 1.4.3 and D-PHY Transmitter IP version 1.7.2: • Figure 3.1. rx_dphy IP Creation in Lattice Radiant Software • Figure 3.2. rx_dphy IP Creation Continued in Lattice Radiant Software • Figure 3.3. Active Data Detection (CSI-2 Frame Start) • Figure 3.4. End of Active Data Detection (CSI-2 Long Packet) • Figure 3.5. Short Packet Write and Read • Figure 3.6. rx_buffer Write/Read Example with Depth = 512, BD_RD_DLY = 300 • Figure 3.7. rx_buffer FIFO Overflow with Depth = 512, BD_RD_DLY = 1000 • Figure 3.8. rx_buffer Successful Transactions with Depth = 1024, BD_RD_DLY = 1000 • Figure 3.9. TX Bandwidth = 2 x RX Bandwidth with Depth = 1024, BD_RD_DLY = 1000 • Figure 3.10. Short Packet Transaction (NUM_TX_LANE_4, TX_GEAR_8) • Figure 3.11. End of Long Packet Transaction (NUM_RX_LANE_4, RX_GEAR_8 to NUM_TX_LANE_2, TX_GEAR_8) • Figure 3.12. Short Packet Transaction in Non-Continuous Clock Mode • Figure 3.13. Global Sequence in Non-Continuous Clock Mode without KEEP_HS • Figure 3.14. Global Sequence in Non-Continuous Clock Mode with KEEP_HS • Figure 3.15. tx_dphy IP General Parameters Creation in Lattice Radiant Software • Figure 3.16. tx_dphy IP General Parameters Creation Continued in Lattice Radiant

Section	Change Summary
	Software • Figure 3.17. tx_dphy IP Protocol Timing Parameters Creation in Lattice Radiant Software
Design Simulation	Updated below figures as per radiant version 3.1: • Figure 5.1. Script Modification #1 • Figure 5.2. Script Modification #2
Design Package and Project Setup	Updated the clock modes and PLL mode of single-lane and four-lane TX channels.

Revision 1.0, February 2021

Section	Change Summary
All	Initial release.

www.latticesemi.com