Lattice Avant 1-to-N MIPI CSI-2/DSI Duplicator Reference Design User Guide

Reference Design

FPGA-RD-02304-1.0

December 2024

Disclaimers

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

Inclusive Language

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

Contents

Contents 3
Abbreviations in This Document 5
1. Introduction 6
1.1. Quick Facts 6
1.2. Features 6
1.3. Naming Conventions 6
1.3.1. Nomenclature 6
1.3.2. Signal Names 6
2. Directory Structure and Files 7
3. Functional Description 8
3.1. Design Components 8
3.1.1. rx_dphy 8
3.1.2. mipi_parser 12
3.1.3. rx_buffer 13
3.1.4. tx_ctrl 14
3.1.5. tx_dphy_if 15
3.1.6. tx_dphy_wrapper 17
3.2. Clocking Scheme 21
3.3. Reset Scheme 24
4. Reference Design Parameter Description 25
4.1. Synthesis Directives 25
4.2. Simulation Directives 26
5. Signal Description 28
6. Running the Reference Design 29
6.1. Compiling the Reference Design 29
7. Simulating the Reference Design 30
7.1. Simulation Results 32
8. Implementing the Reference Design on Board 34
8.1. Hardware Setup 34
8.2. Programming the Board 35
8.3. Hardware Test Result 36
9. Customizing the Reference Design 38
9.1. Design Directives 38
9.2. Rx:Tx Permutations 38
9.3. 1-to-N MIPI CSI-2/DSI Duplicator Calculator 38
10. Resource Utilization 40
11. Debugging 41
11.1. Debug Tools 41
11.1.1. Reveal Analyzer 41
11.1.2. QuestaSim 41
12. Limitation 42
References 43
Technical Support Assistance 44
Revision History 45

Figures

Figure 2.1. Directory Structure 7
Figure 3.1. 1-to-N MIPI CSI-2/DSI Duplicator with the Avant Reference Design Block Diagram 8
Figure 3.2. rx_dphy IP Parameters in General Tab 9
Figure 3.3. rx_dphy IP Parameters in RX_FIFO Settings Tab 10
Figure 3.4. Active Data Detection (CSI-2 Frame Start) 12
Figure 3.5. End of Active Data Detection (CSI-2 Long Packet) 12
Figure 3.6. Short Packet Write and Read 13
Figure 3.7. rx_buffer Write/Read Example with Depth = 512, BD_RD_DLY = 400 13
Figure 3.8. rx_buffer FIFO Overflow with Depth = 512, BD_RD_DLY = 800 14
Figure 3.9. rx_buffer Successful Transactions with Depth = 1,024, BD_RD_DLY = 1,200 14
Figure 3.10. Tx Bandwidth = 2 x Rx Bandwidth with Depth = 1,024, BD_RD_DLY = 1,200 14
Figure 3.11. Short Packet Transaction (NUM_TX_LANE_4, TX_GEAR_8) 15
Figure 3.12. End of Long Packet Transaction (NUM_RX_LANE_4, RX_GEAR_8 to NUM_TX_LANE_4, TX_GEAR_8) 15
Figure 3.13. Short Packet Transaction in Non-Continuous Clock Mode 16
Figure 3.14. Global Sequence in Non-Continuous Clock Mode without KEEP_HS 16
Figure 3.15. Global Sequence in Non-Continuous Clock Mode with KEEP_HS 17
Figure 3.16. tx_dphy_m IP General Parameters Creation in Lattice Radiant Software 18
Figure 3.17. tx_dphy_s IP General Parameters Creation in Lattice Radiant Software 19
Figure 3.18. tx_dphy IP Protocol Timing Parameters Creation in Lattice Radiant Software 21
Figure 3.19. Clocking Scheme Example in Rx Continuous Clock Mode for Two Tx Channels 22
Figure 3.20. Clocking Scheme Example in Rx Non-Continuous Clock Mode for Two Tx Channesl 23
Figure 3.21. Reference Design Reset Scheme 24
Figure 7.1. Simulation Wizard: Create Simulation Project 30
Figure 7.2. Simulation Wizard: Add and Reorder Simulation Source File 31
Figure 7.3. Simulation Wizard: Select Simulation Top Module 31
Figure 7.4. Simulation Wizard: Summary Page 32
Figure 7.5. Example Simulation Result 1 32
Figure 7.6. Example Simulation Result 2 33
Figure 7.7. Functional Simulation Example of CSI-2 with Four Non-Continuous Clock Mode for Tx Channels 33
Figure 8.1. Lattice Avant 1-to-N CSI-2/DSI Duplicator Reference Design Hardware Setup 34
Figure 8.2. Lattice Avant 1-to-N CSI-2/DSI Duplicator Reference Design Hardware Test Block Diagram 35
Figure 8.3. Lattice Avant 1-to-N CSI-2/DSI Duplicator Reference Design Hardware Test Default HDMI Output Display ... 36
Figure 8.4. Lattice Avant 1-to-N CSI-2/DSI Duplicator Reference Design Hardware Test HDMI Output Display after Pressing SW4 37
Figure 9.1. Configuration Helper Calculator 39

Tables

Table 1.1. Summary of the Reference Design 6
Table 3.1. Reference Design Guidelines and Parameter Settings 10
Table 3.2. tx_dphy_m and tx_dphy_s IP Instances Guidelines and Parameter Settings 19
Table 3.3. Clock Resources in Case of Rx D-PHY with Continuous Clock Mode and Non-Continuous Clock Mode 23
Table 4.1. Synthesis Directives 25
Table 4.2. Simulation Directives 26
Table 5.1. CSI-2/DSI Duplicator Top-Level I/O 28
Table 8.1. User Interface on the Avant-X Versa and Avant-E Evaluation Boards 35
Table 9.1. Rx:Tx Permutations 38
Table 10.1. Resource Utilization 40

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviations	Definition
AP	Application Processors
CSI-2	Camera Serial Interface 2
DDR	Double Data Rate
DE	Data Enable
DSI	Display Serial Interface
DUT	Device Under Test
EBR	Embedded Block RAM
EoTp	End of Transmission Packet
FIFO	First In, First Out
FMC	FPGA Mezzanine Card
FPGA	Field Programmable Gate Array
FPU	Floating Point Unit
FS	Frame Start
GPLL	General Purpose PLL
HDL	Hardware Description Language
HS	High Speed
HSYNC	Horizontal Sync
I/O	Input/Output
ISP	Image Signal Processing
LP	Low Power
LSB	Lease Significant Bit
LUT	Look Up Table
MIPI	Mobile Industry Processor Interface
MSB	Most Significant Bit
OEM	Original Equipment Manufacturer
PLL	Phase Locked Loop
Rx	Receiver
STA	Static Timing Analysis
Tx	Transmitter
USB	Universal Serial Bus
VSS	Vsync Start
VSYNC	Vertical Sync
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 Avant™ 1-to-N MIPI® CSI-2®/DSI Duplicator reference design can be configured for one to four duplicated outputs. The Lattice Radiant™ MIPI D-PHY Soft IPs can be configured as a MIPI transmitter or receiver utilizing the general DDR modules.

1.1. Quick Facts

Download the reference design files from the 1 to N MIPI CSI-2/DSI Duplicator Reference Design web page.

Table 1.1. Summary of the Reference Design

General	Target Devices	LAV-AT-X70
	Source Code Format	Verilog
Simulation	Functional Simulation	Performed
	Timing Simulation	Performed
	Test Bench	Available
	Test Bench Format	Verilog
Software Requirements	Software Tool and Version	Lattice Radiant 2024.1
	IP Version (if applicable)	CSI-2/DSI Receiver v1.7.0 CSI-2/DSI Transmitter v2.0.0 PLL v2.5.0
Hardware Requirements	Board	Avant-X Versa Board
	Cable	USB to mini-USB cable for programming purpose

1.2. Features

Key features of the Lattice Avant 1-to-N MIPI CSI-2/DSI Duplicator reference design include:

- 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.
- Maximum Rx bandwidth is 1.2 Gb/s per lane using the 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 1.2 Gb/s per lane using the D-PHY Soft IP.
- Non-continuous clock mode on Rx channels is possible as long as a continuous clock is provided internally or from an external source.
- Non-continuous or continuous clock mode is possible for Tx channels.

1.3. Naming Conventions

1.3.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.3.2. Signal Names

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

2. Directory Structure and Files

Figure 2.1 shows the directory structure of the Lattice Avant 1-to-N MIPI CSI-2/DSI Duplicator reference design.

```
.
└── avant_mipi_1toN_duplicator_rd/
    ├── mipi_dup_rd_avant.rdf                          - Lattice Radiant design file
    ├── docs/                                          - Contains IP configuration helper calculator
    │   └── mipi_dup_RD_Avant.xlsx
    ├── demo_bitfiles /                                - Contains pre-compiled bitstream files for hardware evaluation
    │   ├── mipi_dup_1to2_versa.bit
    │   └── mipi2hdmi_2to1_eval.bit
    ├── impl1                                          - Contains implementation files and other Radiant generated files
    ├── source/                                        - Contains verilog top level design, design modules and synthesis_directive file
    │   └── verilog/
    │       └── avant/
    │           ├── mipi_dup_top.v
    │           ├── synthesis_directives.v
    │           ├── mipi_duplicator_avant.v
    │           ├── mipi_parser.v
    │           ├── reset_controller.v
    │           ├── rx_buffer.v
    │           ├── tx_ctrl.v
    │           ├── tx_dphy_if.v
    │           ├── tx_dphy_wrapper.v
    │           └── cam_i2c/
    │               ├── camera_i2c_master.v
    │               ├── camera_i2c_master_controller.v
    │               ├── camera_i2c_top.v
    │               ├── imx258_bare_2k.mem
    │               └── imx258_bare_4k.mem
    ├── testbench/                                     - Contains simulation_directives and other simulation files including testbench
    │   └── verilog/
    │       ├── mipi_duplicator_avant_tb.v
    │       ├── simulation_directives.v
    │       ├── bus_driver.v
    │       ├── clk_driver.v
    │       ├── csi2_checker_dup.v
    │       ├── csi2_model.v
    │       ├── dphy_lp_hs_checker.v
    │       ├── dphy_monitor.v
    │       ├── dphy_scoreboard.v
    │       ├── dsi_checker_dup.v
    │       └── dsi_model.v
    ├── int_gpll                                       - Contains Radiant IP file used in the design
    ├── int_osc                                        - Contains Radiant IP file used in the design
    ├── rx_dphy                                        - Contains Radiant IP file used in the design
    ├── tx_dphy_m                                      - Contains Radiant IP file used in the design
    ├── tx_dphy_s                                      - Contains Radiant IP file used in the design
    ├── rom_imx258_2k                                  - Contains Radiant IP file used in the design
    └── rom_imx258_4k                                  - Contains Radiant IP file used in the design
```

Figure 2.1. Directory Structure

3. Functional Description

Figure 3.1 shows the block level diagram of the 1-to-N MIPI CSI-2/DSI Duplicator with the Avant reference design up to four Tx channels. The int_gpll is used to generate a required high-speed clock and feed it to multiple D-PHY Tx IPs.

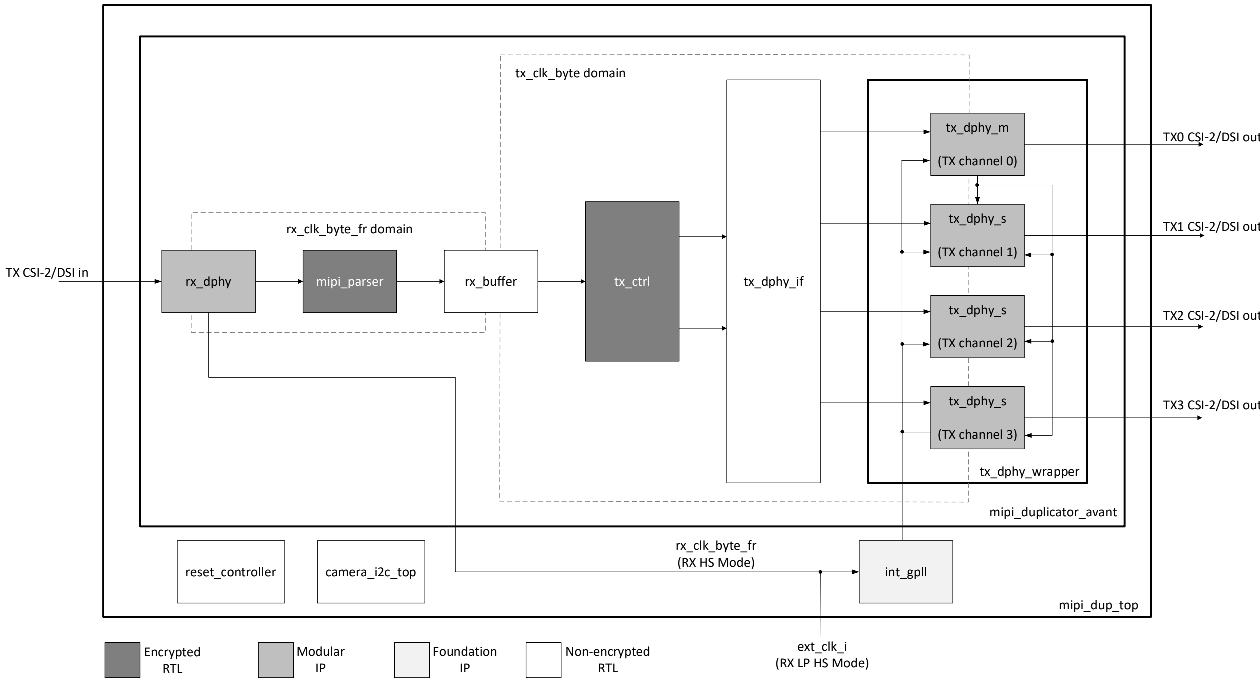

Figure 3.1. 1-to-N MIPI CSI-2/DSI Duplicator with the Avant Reference Design Block Diagram

3.1. Design Components

The 1-to-N MIPI CSI-2/DSI Duplicator with the Avant reference design includes the following blocks:

- rx_dphy
- mipi_parser
- rx_buffer
- tx_ctrl
- tx_dphy_if
- tx_dphy_wrapper

3.1.1. rx_dphy

This module must be created for the Rx channel according to the channel conditions, such as the number of lanes, bandwidth, and others. Figure 3.2 shows an example of IP interface settings in Lattice Radiant software for the CSI-2/DSI D-PHY Receiver Submodule IP version 1.7.0. Refer to the CSI-2/DSI D-PHY Receiver Submodule (Rx IP Core) User Guide (FPGA-IPUG-02081) for details.

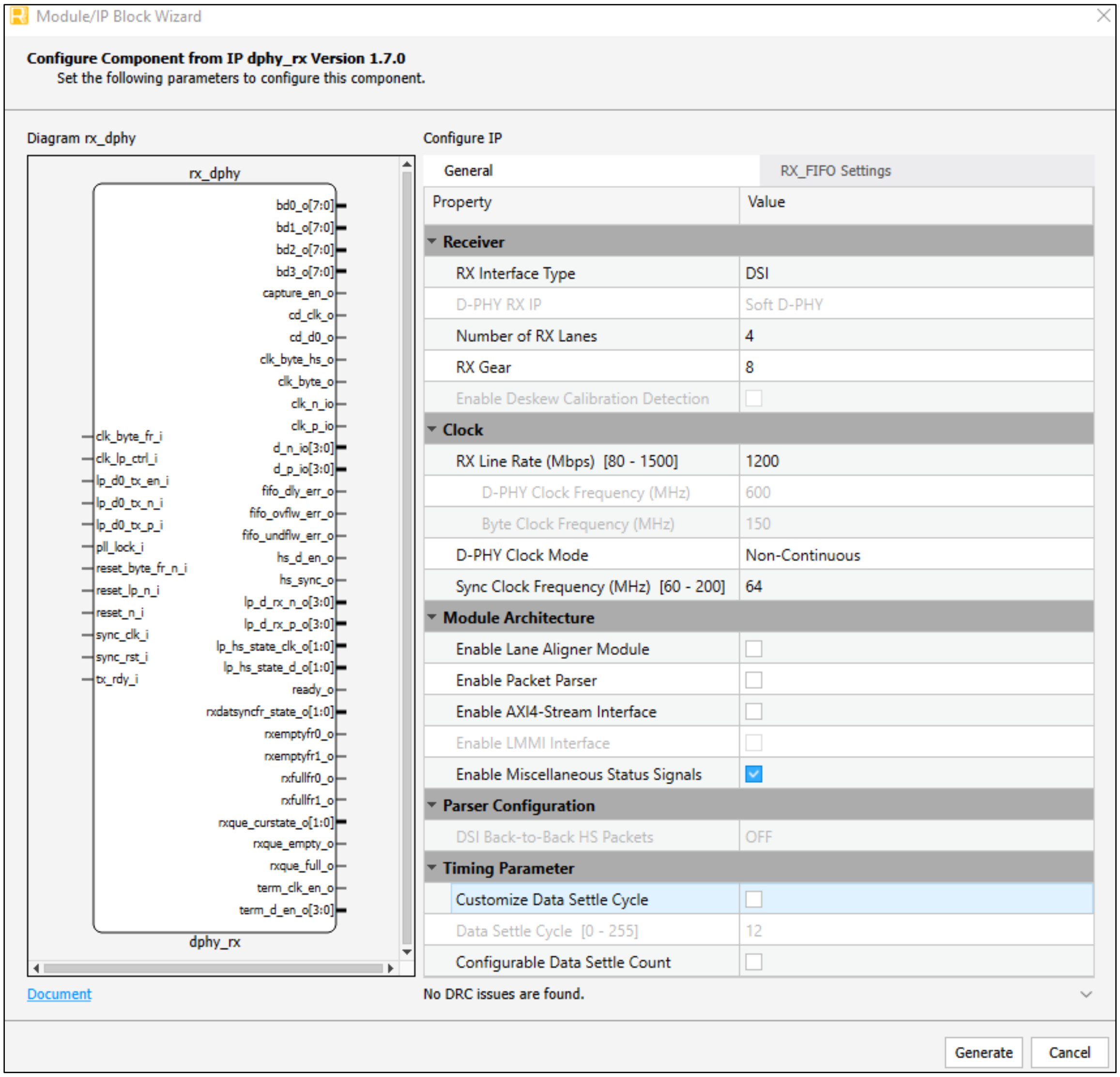

Figure 3.2. rx_dphy IP Parameters in General Tab

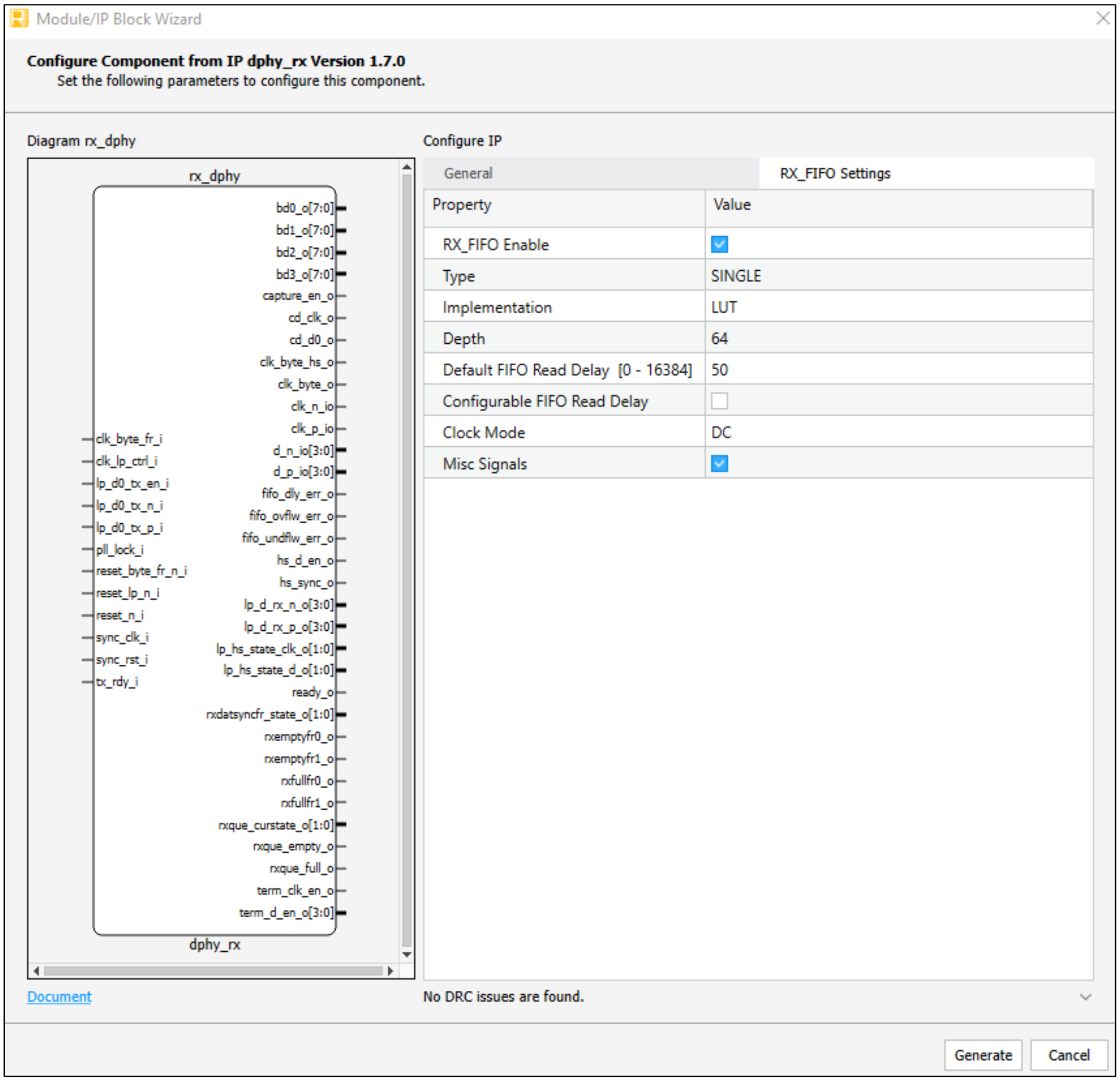

Figure 3.3. rx_dphy IP Parameters in RX_FIFO Settings Tab

Table 3.1 shows the guidelines and parameter settings required for this reference design.

Table 3.1. Reference Design Guidelines and Parameter Settings

Parameter	Description
Rx Interface Type	Select CSI-2 or DSI.
Number of Rx Lanes	Set according to the channel configuration. The value must match the NUM_RX_LANE_* setting in synthesis_directives.v.
Rx Gear	Select 8.
Rx Line Rate (Mb/s)	Set the required line rate. For the Avant devices with CSI-2/DSI Receiver v1.7.0 IP, the maximum line rate is 1,200 Mb/s without the dynamic clock-to-data calibration of soft D-PHY.
D-PHY Clock Mode	Select Continuous or Non-continuous. Must match the RX_CLK_MODE_* setting (Continuous = HS_ONLY, Non-continuous = HS_LP) in synthesis_directive.v.
Sync Clock Frequency	Use the default value of 64 MHz from the oscillator.
Enable Lane Aligner Module	Always disable (uncheck) for Rx D-PHY having two-lane and four-lane configurations. This option is not available for Rx D-PHY with one-lane configuration.
Enable Packet Parser	Always disable (uncheck) for D-PHY channels.
Enable AXI Stream Interface	Always disable (uncheck).

Parameter	Description
Enable Miscellaneous Status Signals	Always enable (check) for SOFT DPHY channels.
DSI Back-to-Back HS Packets	Always disable (uncheck).
Customize Data Settle Cycle	It is recommended to use the suggested value. Try reducing this value one by one when Rx D-PHY is not working as expected.
Configurable Data Settle Count	Always disable (uncheck).
Rx FIFO Enable	Set according to the channel configuration. It can be enabled or disabled.
Type	Set Rx FIFO type according to the channel configuration.
Implementation	Set Rx FIFO implementation according to the channel configuration.
Depth	Set Rx FIFO depth according to the channel configuration. Minimum value of 16 is required.
Default FIFO Read Delay	Set according to the channel configuration. Minimum value of 1 is required.
Clock mode	Always set to DC (Dual clock) for Rx D-PHY.
Miscellaneous Signals	Always enable (check) FIFO miscellaneous signals for D-PHY channels.

Note: After configuring all the required parameters, click **Generate**, which is shown at the bottom left corner of the user interface.

3.1.1.1. Rx FIFO in Non-Continuous Clock Mode

Rx FIFO is useful especially in the non-continuous clock mode and when the continuous byte clock cannot have the exact same frequency as the non-continuous byte clock used in the D-PHY Rx IP. It resides before the word aligner.

In the non-continuous clock mode case, the Rx FIFO configuration depends on the relationship between the non-continuous byte clock in the D-PHY Rx IP and the continuous byte clock, which is most likely generated by the 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. 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 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 the 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 at the end of HS transmission.

Unknown Continuous Byte Clock and Non-Continuous Byte Clock Relationship

The frequency relationship is unknown when it is unclear which clock is faster and when the continuous byte clock is within the certain range against the non-continuous byte clock. For example, if the two clocks come from different clock sources with ppm tolerance. 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 two-lane Gear 8 RAW10 with 1,920 horizontal pixels.

Payload byte count = (1,920 × 10) ÷ 8 = 2,400

So, each lane takes 1,200 bytes.

1,200 × 500 ppm = 0.6, which means small FIFO with middle point read delay such as FIFO depth = 16 with read delay of 8-byte clock cycles works fine. Using LUT is preferable as long as the 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, such as whether it contains line start/end short packet or 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 the total number of EBR used in the device do not exceed the maximum number of EBRs available in the Avant device.

3.1.1.2. Rx FIFO in Continuous Clock Mode

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

3.1.2. mipi_parser

This module handles the 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 3.4 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.

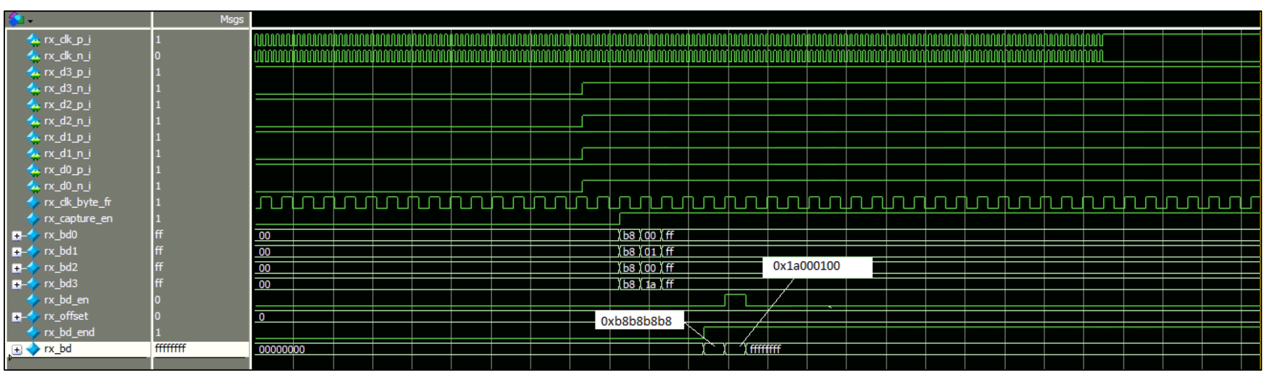

Figure 3.4. Active Data Detection (CSI-2 Frame Start)

Figure 3.5 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 = 2, which means the byte offset of 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.

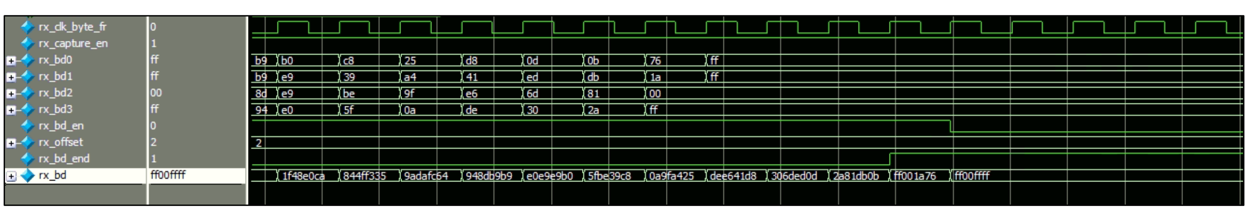

Figure 3.5. End of Active Data Detection (CSI-2 Long Packet)

3.1.3. rx_buffer

This module contains a dual clock FIFO using EBR to store valid packet data. Buffer depth is 512, 1,024, 2,048, or 4,096 and data width is 32. This module itself with ready flag assertion reads out the first FIFO data. The FIFO has extra data width of 32 to contain the data end flag and offset data. Figure 3.6 shows the example of a 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 first FIFO data is automatically read out by this module so that the first 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 a 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 the Tx channel. This delay is expected to be ~1 µs – 1.5 µs in case that both clock and data lanes require 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 the same as shown in Table 9.1, 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, which is 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 avoid FIFO overflow when tx0_clk_byte is slower than rx_clk_byte_fr.
If you are not clear on which clock is faster but expect some tolerance, then setting it close to the middle value makes sense. The minimum buffer depth is 512 and that requires two EBRs (for data width of 32) or four EBRs (for data width of 64). This depth is expandable to 1,024, 2,048, or 4,096 as long as the total number of EBR does not exceed the maximum number of EBRs available in the Avant device.

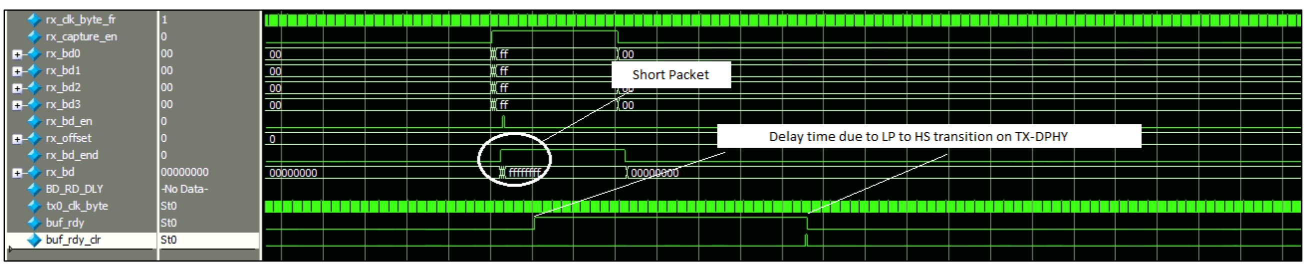

Figure 3.6. Short Packet Write and Read

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 2,400, which means it takes 1,200 rx_clk_byte_fr cycles to write all long packet payload data to FIFO. tx0_clk_byte exactly matches the derived clock ratio specified in Table 9.1. In case of BD_RD_DLY = 400, FIFO read (buf_re = 1) begins while FIFO has some room to be filled up. In case of BD_RD_DLY = 800, data is corrupted due to FIFO overflown shown by a spike of rx_fifo_full in Figure 3.8, inside a red box.

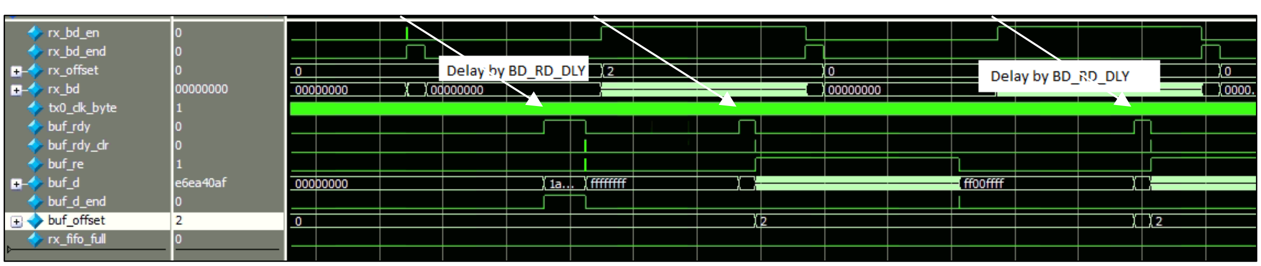

Figure 3.7. rx_buffer Write/Read Example with Depth = 512, BD_RD_DLY = 400

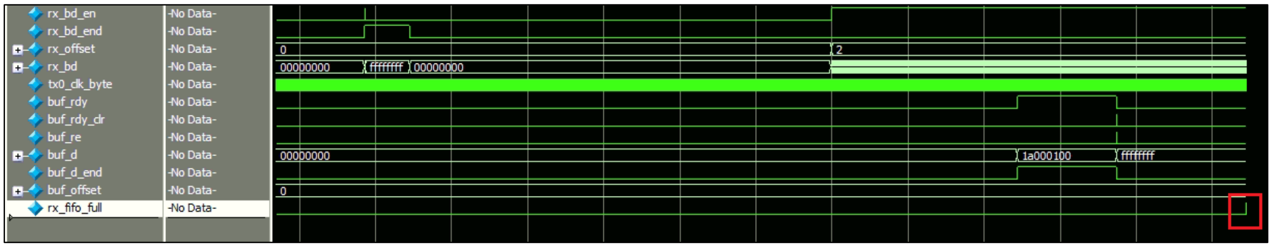

Figure 3.8. rx_buffer FIFO Overflow with Depth = 512, BD_RD_DLY = 800

The safest way to avoid FIFO overflow in case of total Tx bandwidth > Rx total bandwidth, that is the byte clock ratio value is larger than the ratio shown in Table 9.1, is to set the FIFO depth larger than the size of long packet data. Figure 3.9 shows the successful transaction of setting the FIFO depth to 1,024 with BD_RD_DLY = 1,200. 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 1,024 is deep enough to store the whole one-line data (1,024 × 4 (bytes) > (4 (Header) + 2,400 (payload) + 2 (CRC)) and FIFO overflow does not occur.

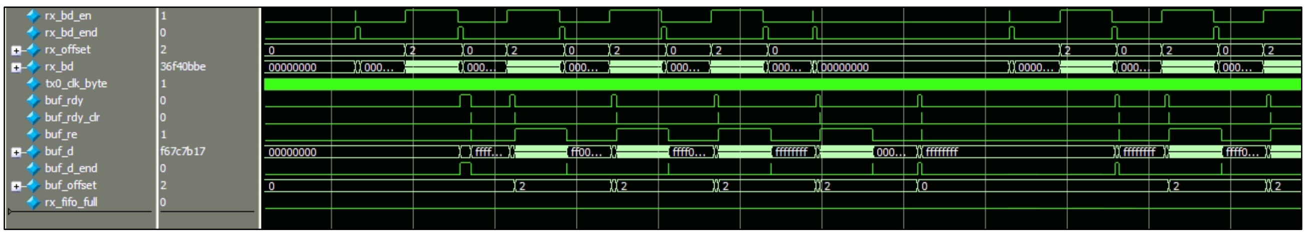

Figure 3.9. rx_buffer Successful Transactions with Depth = 1,024, BD_RD_DLY = 1,200

Figure 3.10 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 the Tx channel. This method, that is 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 the Rx bandwidth and does not exceed the maximum lane bandwidth of 1.2 Gb/s 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 be applied, and the Tx bandwidth has to be exactly the same (or close enough) to avoid FIFO overflow/underflow.

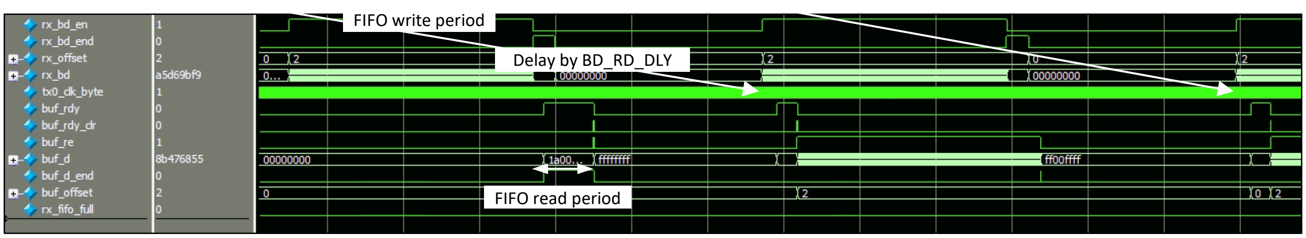

Figure 3.10. Tx Bandwidth = 2 x Rx Bandwidth with Depth = 1,024, BD_RD_DLY = 1,200

3.1.4. 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 3.11 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).

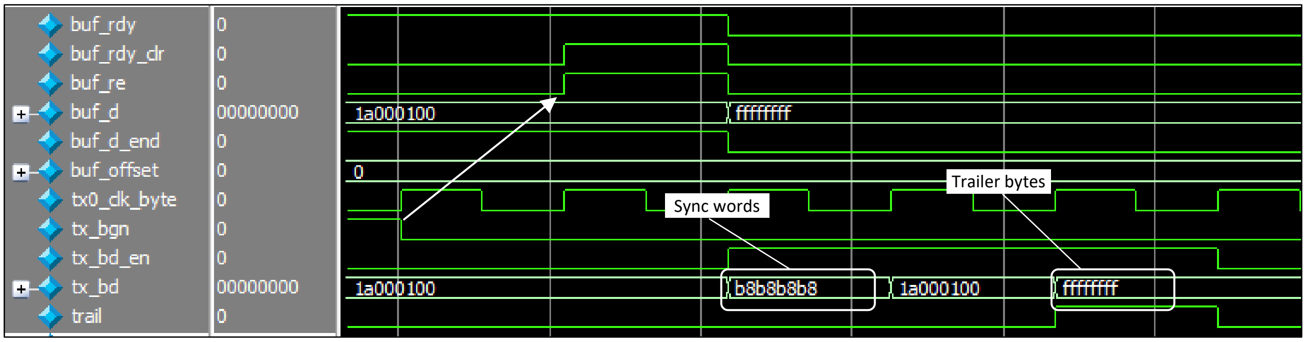

Figure 3.11. Short Packet Transaction (NUM_TX_LANE_4, TX_GEAR_8)

Figure 3.12 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.

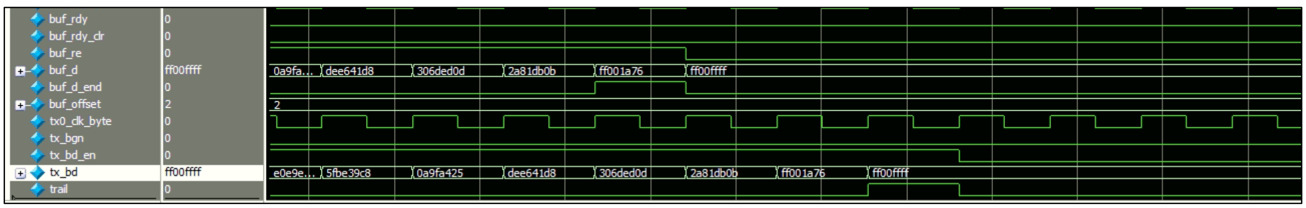

Figure 3.12. End of Long Packet Transaction (NUM_RX_LANE_4, RX_GEAR_8 to NUM_TX_LANE_4, TX_GEAR_8)

When Tx D-PHY is set to the 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.

3.1.5. 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 3.13 shows an example of a short packet transaction in the non-continuous clock mode. Only data lane 0 is shown in Figure 3.14.

For more information, refer to the sequence below:

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.

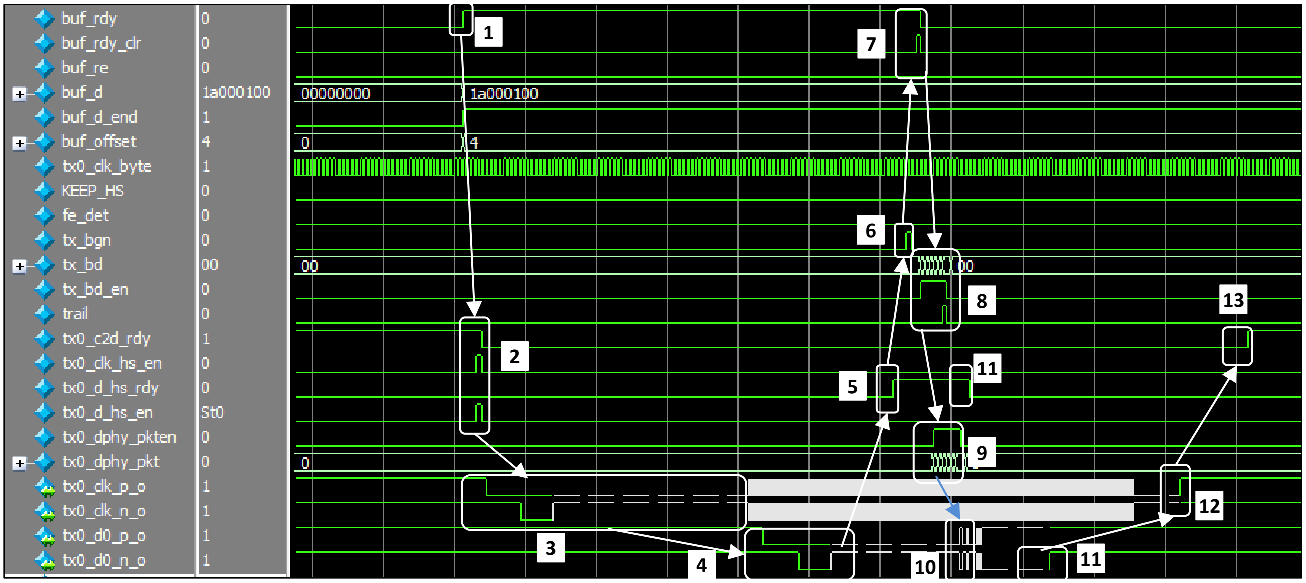

Figure 3.13. Short Packet Transaction in Non-Continuous Clock Mode

For CSI-2 in the non-continuous clock mode, it is possible to keep the clock lane in HS mode during the horizontal blanking periods by defining KEEP_HS in synthesis_directives.v. Figure 3.14 shows the global sequence without defining KEEP_HS and Figure 3.15 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.

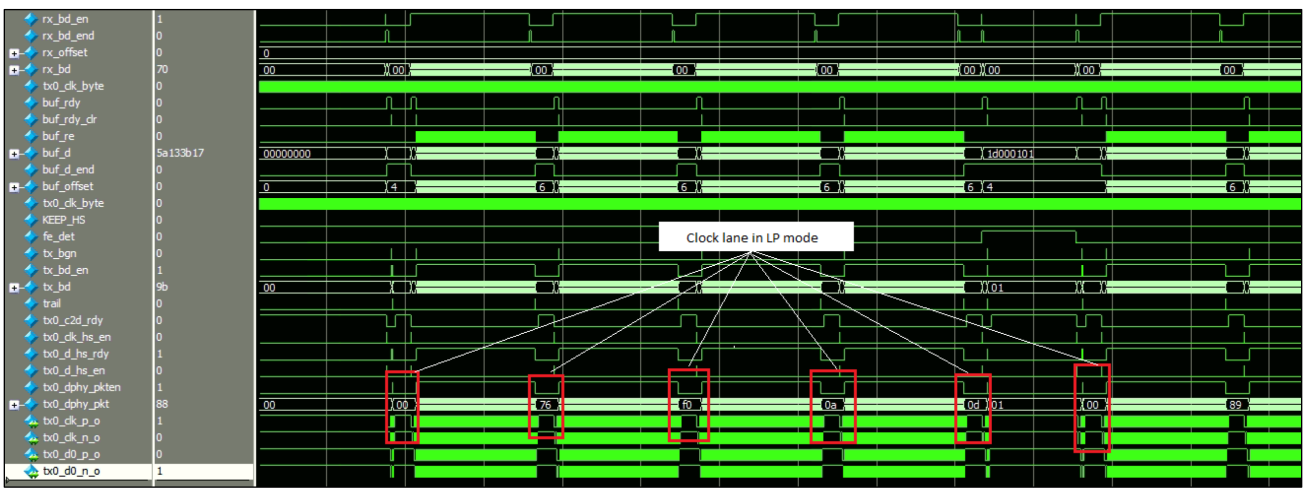

Figure 3.14. Global Sequence in Non-Continuous Clock Mode without KEEP_HS

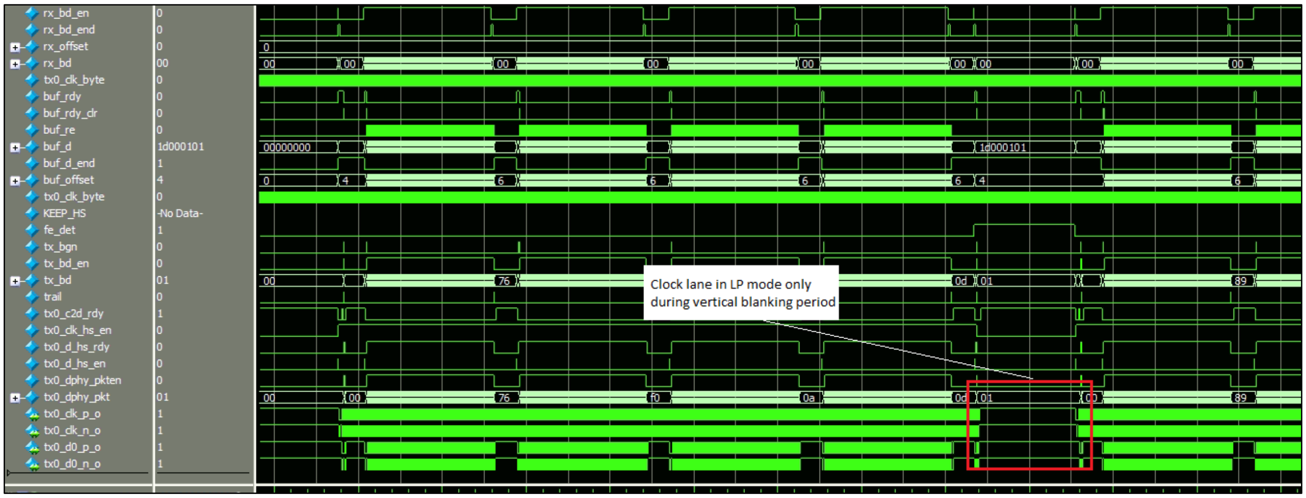

Figure 3.15. Global Sequence in Non-Continuous Clock Mode with KEEP_HS

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 the transactions.

3.1.6. tx_dphy_wrapper

This module instantiates multiple tx_dphy according to the number of Tx channels defined in the synthesis_directives.v. If there is more than one Tx channel, the module will instantiate one primary dphy_tx (dphy_tx_m) that is used to generate the ECLK elements as per the Avant device requirement. The remaining dphy_tx (dphy_tx_s) instances utilize the ECLK elements from the primary dphy_tx. You must ensure all dphy_tx instances have the same configuration except for the Enable Edge Clock Synchronizer and Divider attributes in the IP for it to work properly.

Tx D-PHY module takes the byte data and outputs DSI/CSI-2 data after serialization in the HS mode. It is recommended to set the module name to tx_dphy_m (primary) and tx_dphy_s (secondary) so that you do not need to modify the instance name of this IP in the design as well as the simulation setup file. Otherwise, you need to modify the names accordingly.

You must update this module according to the channel conditions, such as number of lanes, bandwidth, and others. Figure 3.16 shows an example of tx_dphy_m IP interface setting in the Lattice Radiant software for the CSI-2/DSI D-PHY Transmitter Submodule IP. Refer to the CSI-2/DSI D-PHY Transmitter Submodule (Tx IP Core) User Guide (FPGA-IPUG-02080) for details.

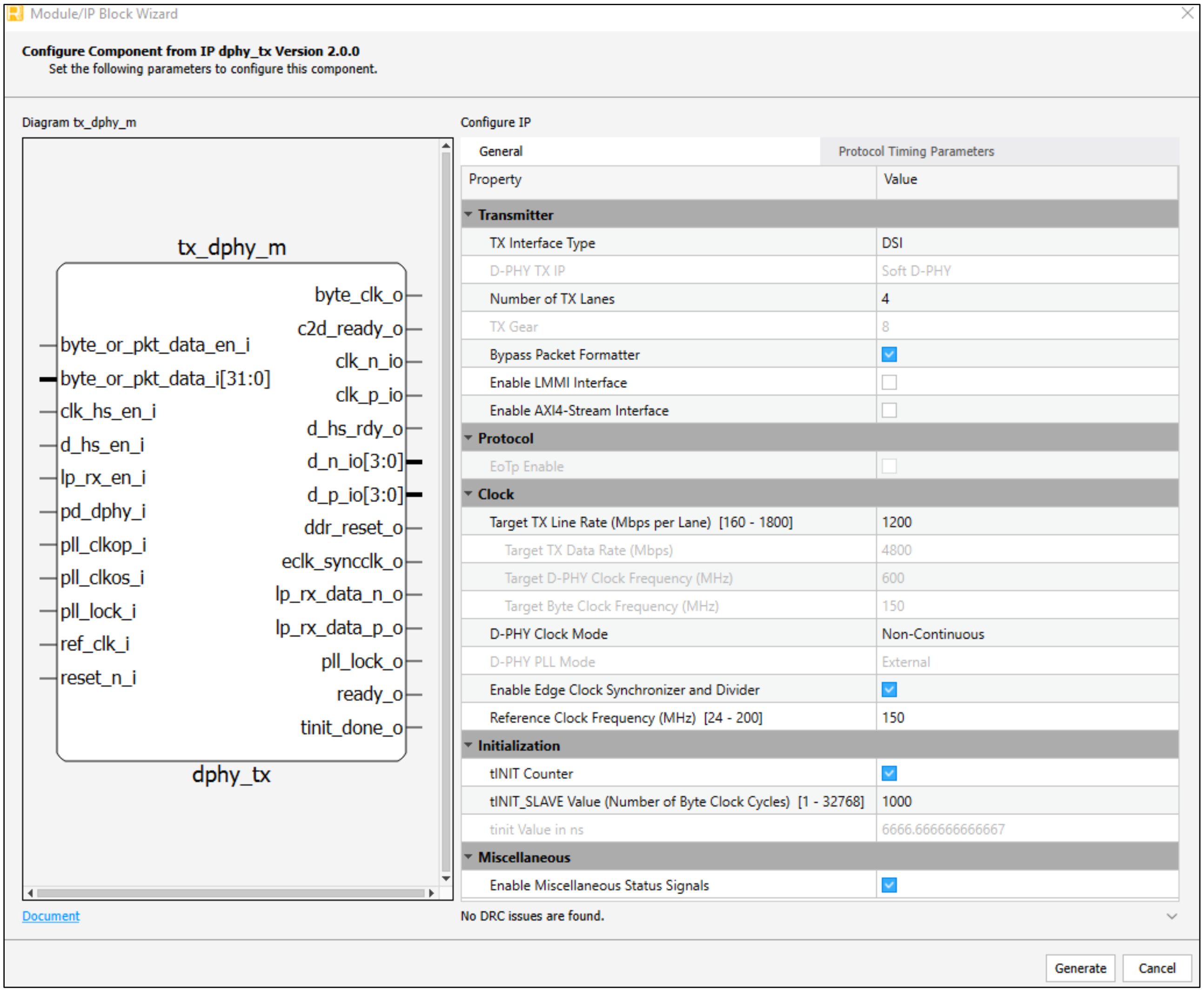

Figure 3.16. tx_dphy_m IP General Parameters Creation in Lattice Radiant Software

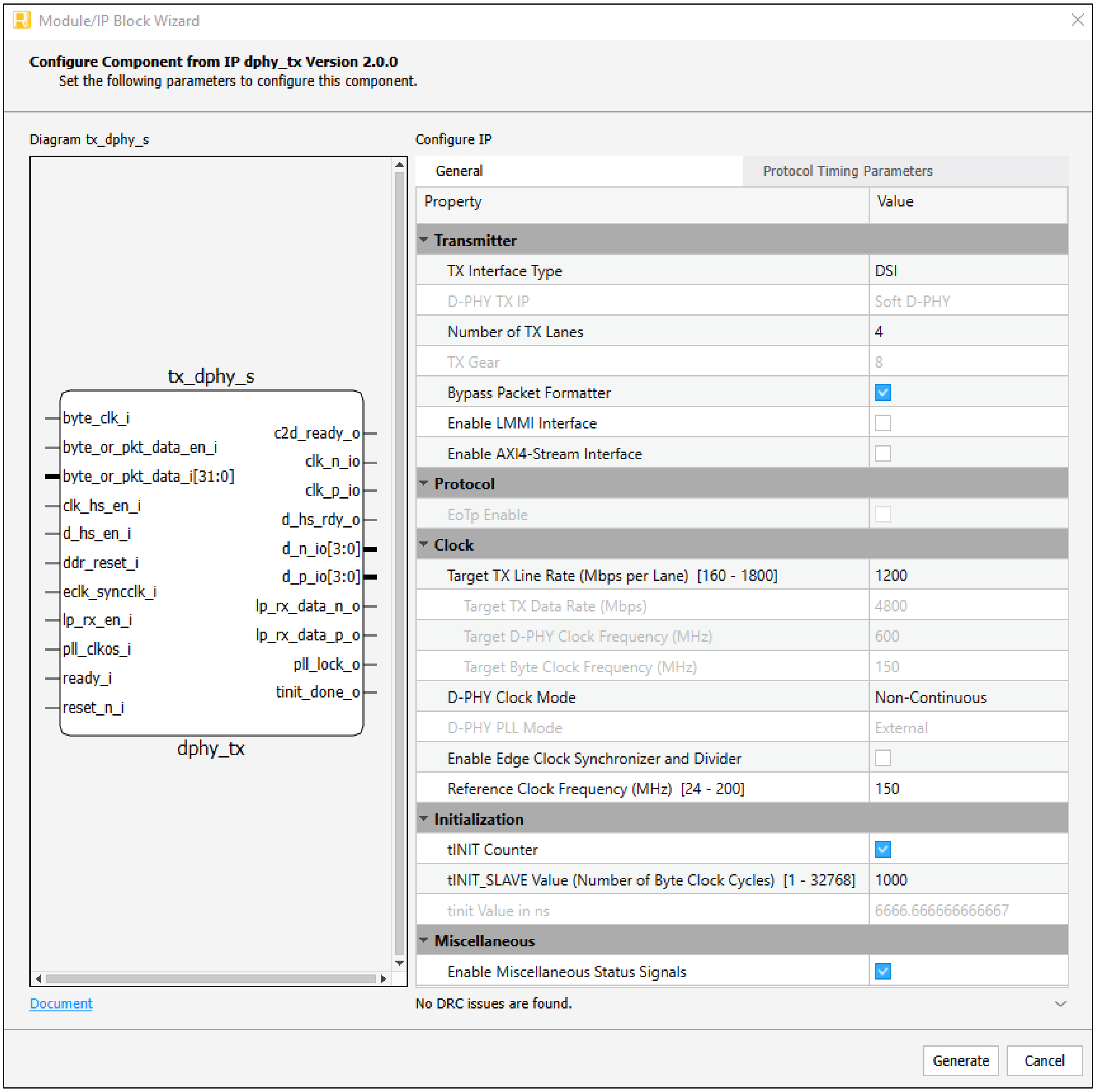

Figure 3.17. tx_dphy_s IP General Parameters Creation in Lattice Radiant Software

Table 3.2 shows the guidelines and General parameter settings required for tx_dphy_m and tx_dphy_s IP instances.

Table 3.2. tx_dphy_m and tx_dphy_s IP Instances Guidelines and Parameter Settings

Parameter	Description
Tx Interface Type	Select CSI-2 or DSI.
DPHY Tx IP	The Avant devices support only Soft DPHY type.
Number of Tx Lanes	Set according to the channel configuration. Must match NUM_TX_LANE_* setting in synthesis_directives.v.
Tx Gear	Set according to Tx Line Rate.
Bypass Packet Formatter	Always enable (check).
Enable LMMI Interface	Always disable (uncheck).

Parameter	Description
Enable AXI Stream Interface	Always disable (uncheck).
EoTP Enable	Available when Tx Interface Type = DSI. Always disable (uncheck).
Enable Frame Number Increment in Packet Formatter	Always disable (uncheck).
Frame Number MAX Value Increment in Packet Formatter [1 - 255]	The default value is 1.
Enable Line Number Increment in Packet Formatter	Always disable (uncheck).
Extended Virtual Channel ID	Always disable (uncheck).
Target Tx Line Rate	Set according to the channel configuration. The Tx and Rx bandwidths must be the same: Rx Line Rate * Number of Rx Lane = Tx Line Rate * Number of Tx Lane
D-PHY Clock Mode	Set according to the channel configuration. Select Continuous when the horizontal blanking period is short in DSI. In CSI-2 case, using Non-Continuous and defining KEEP_HS could be an option for short horizontal blanking.
D-PHY PLL Mode	Continuous or Non-Continuous. Set according to the channel configuration.
Enable Edge Clock Synchronizer and Divider	Check for primary tx_dphy_m instance, uncheck for secondary tx_dphy_s instance.
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).
tINIT_SLAVE Value	Recommend to set this to 1,000, which is the default value.
Enable Miscellaneous Status Signals	Always enable (check).

Note: After configuring all required General parameters, click **Generate**, which is shown at the bottom left corner of the user interface.

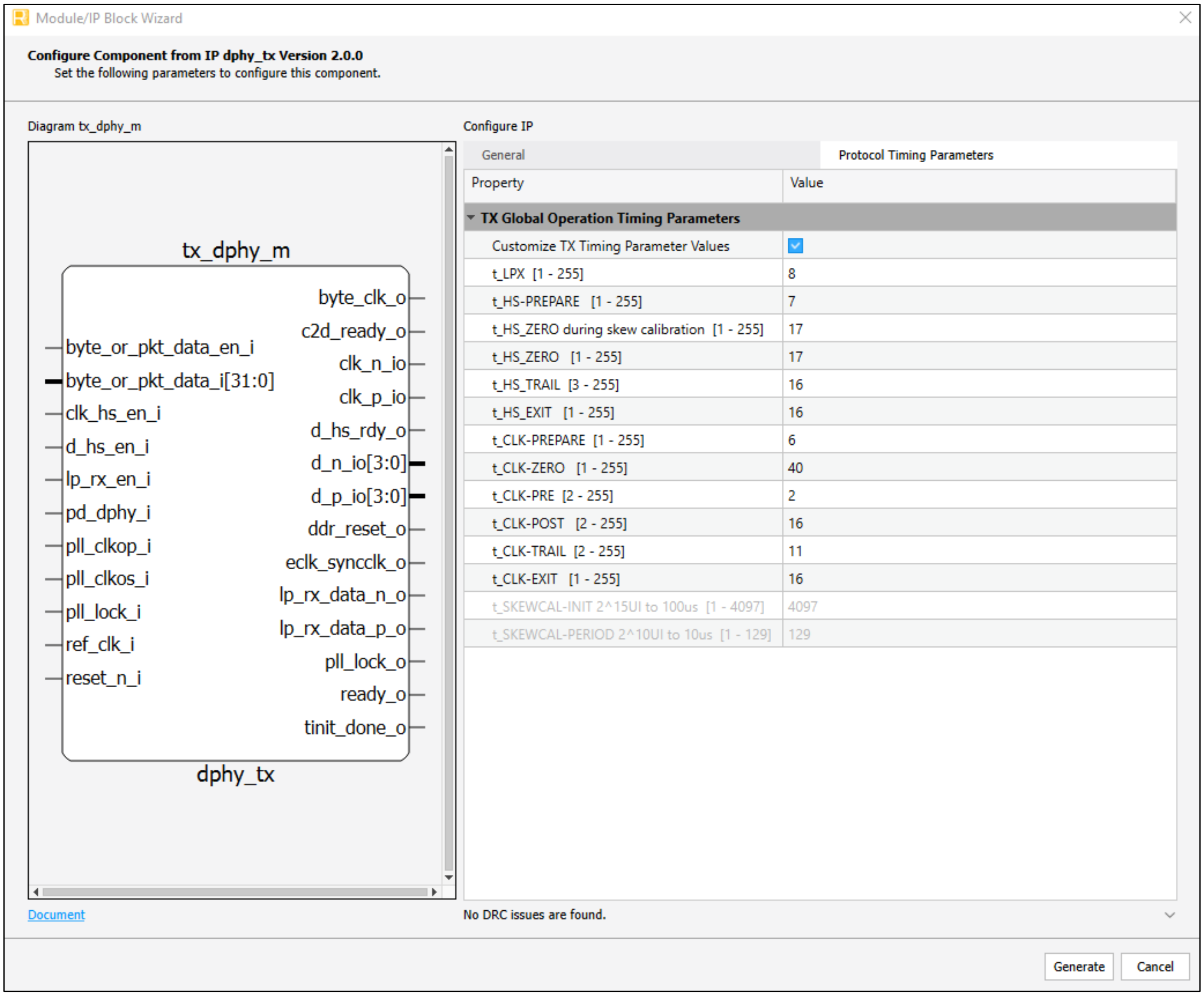

Figure 3.18. tx_dphy IP Protocol Timing Parameters Creation in Lattice Radiant Software

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

- While generating a Tx D-PHY IP in the Lattice Radiant software, the Protocol timing parameters are set to appropriate values according to the Tx D-PHY General Clock settings as shown in Figure 3.18.
- If you need to customize these values while creating Tx D-PHY IP, after configuring all General parameters, navigate to the Protocol Timing Parameters window, and enable (check) the Customize Tx Timing Parameter Value option. After changing to desired parameter value, click **Generate**, which is shown at the bottom left corner of the user interface.
- For any configuration if D-PHY LP00 + HS00 transition Period is less than 300 ns, 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, click **Generate**, which is shown at the bottom left corner of the user interface.
- For more information, refer to the CSI-2/DSI D-PHY Transmitter Submodule (Tx IP Core) User Guide (FPGA-IPUG-02080).

3.2. Clocking Scheme

Figure 3.19 shows a clocking scheme example in Rx continuous clock mode for two Tx channels with external PLL configuration. The int_gpll needs to generate two high-speed clocks, which are pll_clkop and pll_clkos. pll_clkos has the same frequency as pll_clkop but it is 90-degree phase shifted from pll_clkop. pll_clkop and pll_clkos are shared across tx_dphy_m and tx_dphy_s since they are assumed to have the same configuration. It is also assumed that tx_dphy_s is placed at the same or adjacent bank as tx_dphy_m to save resource and to allow for ECLK sharing between those two. By doing this, both data interfaces of tx_dphy_m and tx_dphy_s are synchronized to the byte_clk of the tx_dphy_tx_m.

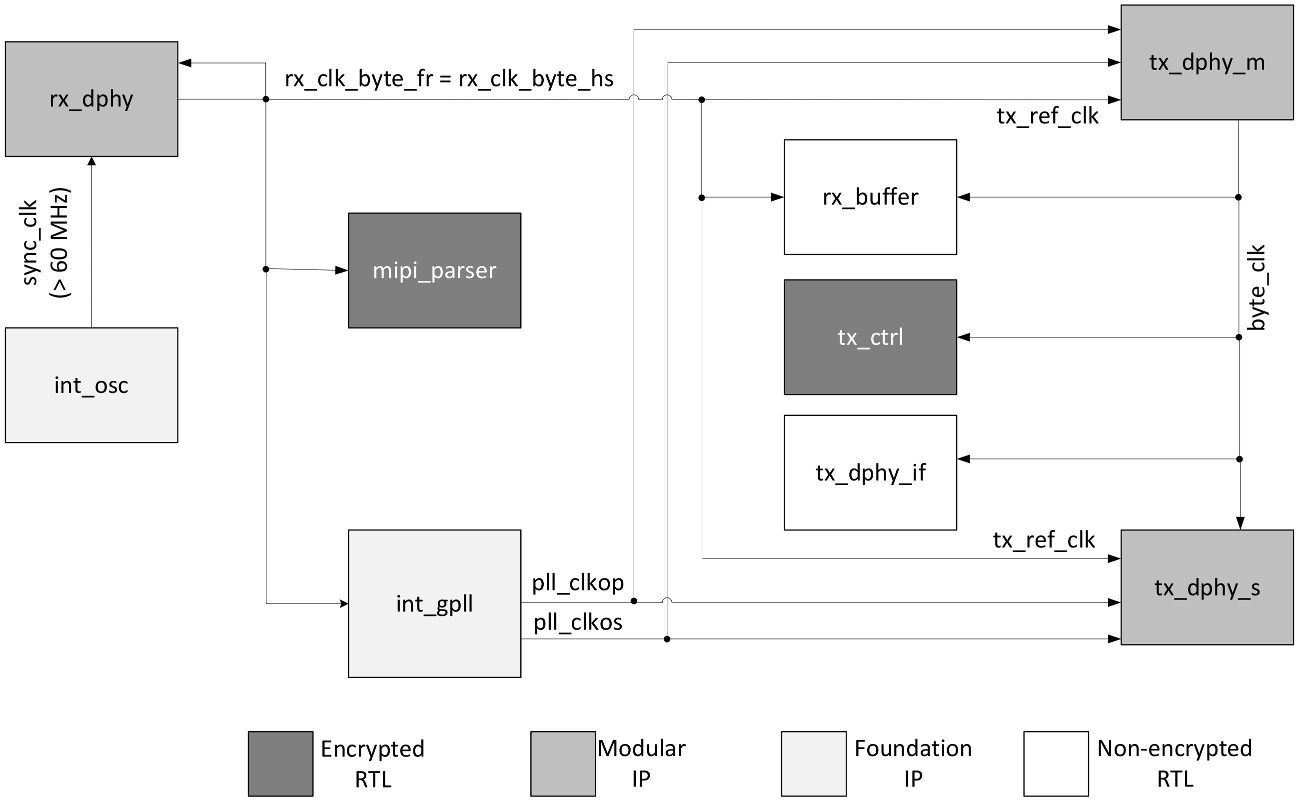

Figure 3.19. Clocking Scheme Example in Rx Continuous Clock Mode for Two Tx Channels

Figure 3.20 shows a clocking scheme example in Rx non-continuous clock mode for two Tx channels with external PLL configuration. Similar to Rx Continuous clock mode, the sync_clk is generated by the oscillator. The int_gpll is used to generate continuous Rx byte clock and D-PHY Tx high-speed clock, which are pll_clkop and pll_clkos. pll_clkos has the same frequency as pll_clkop but it is 90-degree phase shifted from pll_clkop. pll_clkop and pll_clkos are shared across tx_dphy_m and tx_dphy_s since they are assumed to have the same configuration. It is also assumed that tx_dphy_s is placed at the same or adjacent bank as tx_dphy_m to save resource and to allow for ECLK sharing between those two. By doing this, both data interfaces of tx_dphy_m and tx_dphy_s are synchronized to the byte_clk of the tx_dphy_tx_m.

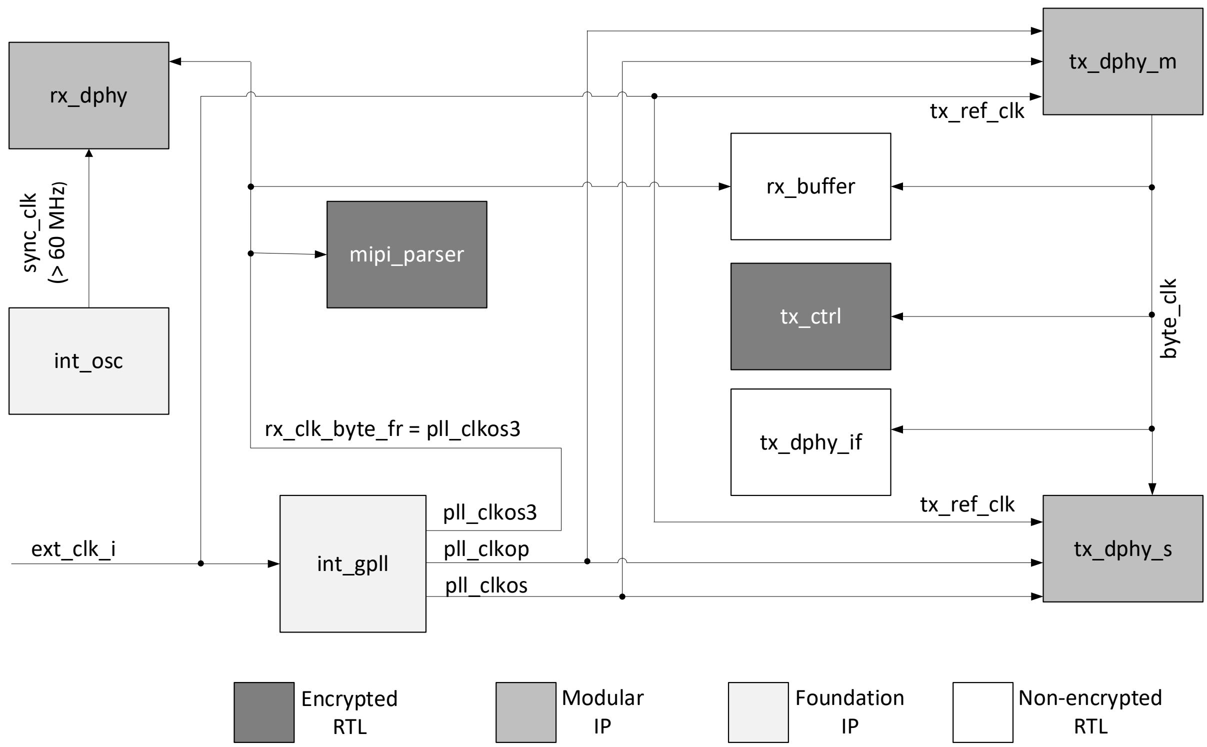

Figure 3.20. Clocking Scheme Example in Rx Non-Continuous Clock Mode for Two Tx Channesl

Also note that the reference design has typical clock assignments to rx_clk_byte_fr and rx_clk_lp_ctrl according to the clock mode on the Rx channel. The rx_clk_lp_ctrl frequency, which is the clock signal for LP and HS mode control module for clock lane, can be different from the rx_clk_byte_fr frequency, which is the continuous byte clock for the Rx channel. It is recommended for the two frequencies to be the same to save the primary clock tree resources.

On the Tx side, using continuous or non-continuous clock mode does not affect the number of necessary clock trees. Always use two clock tree per Tx channel for clkop and clkos. In general, it is recommended to use the same clock mode as the Rx side. Special care is needed when using non-continuous mode on Tx side while the Rx side operates in continuous mode since LP-HS transition time on the Rx side might not be long enough to allow Tx side to make LP-HS transitions, including the clock lane, which leads to FIFO overflow. To feed a clock to the 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 operate within the range of 24–200 MHz.

Table 3.3 shows the typical clock resource for different configurations, serving as a guideline for your clocking scheme configuration.

Table 3.3. Clock Resources in Case of Rx D-PHY with Continuous Clock Mode and Non-Continuous Clock Mode

Rx D-PHY		
Maximum Rx Lane Bandwidth	1,200 Mb/s	
D-PHY Clock Mode	HS_ONLY (Continuous Clock)	HS_LP (Non-Continuous Clock)
Clk_lp_ctrl_i	NA	OSC
Sync_clk_i	OSC	OSC
Use ext_clk_i	No	Yes
GPLL clki_i	rx_clk_byte_fr	ext_clk_i
Rx_clk_byte_fr	rx_clk_byte_hs	pll_clkos3
Tx D-PHY		
Maximum Tx Lane Bandwidth	1,200 Mb/s	
D-PHY Clock Mode	HS_ONLY (Continuous Clock)	HS_LP (Non-Continuous Clock)
Tx_ref_clk	rx_clk_byte_fr	ext_clk_i

3.3. Reset Scheme

The system-level reset is routed to the reset_n_i pin of the top-level module as an active-low reset. Asserting this reset asynchronously resets int_gpll. The reset_n_i pin also serves as an input to the reset_controller module, which controls the reset timing for the camera sensor and its I2C configuration, as well as connected downstream devices. This design also uses the reset signal to control the power-down signal for Tx D-PHY. The de-assertion of the reset_n_i signal asserts the tx_pd_dphy signal.

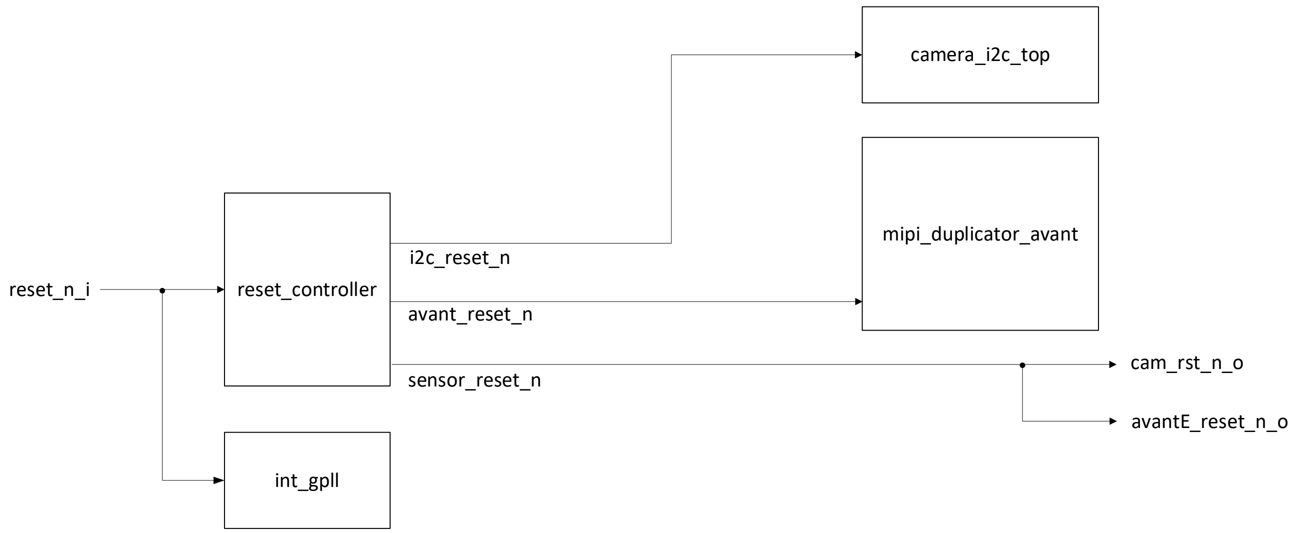

Figure 3.21. Reference Design Reset Scheme

4. Reference Design Parameter Description

The two directive files for this reference design are as follows:

- 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 Components section.

These directives are used for both synthesis and simulation. As shown in Table 4.1 and Table 4.2, some parameter selections are restricted by other parameter settings.

4.1. Synthesis Directives

Table 4.1 shows the synthesis directives that affect this reference design.

Table 4.1. Synthesis Directives

Category	Directives	Descriptions
GPLL	USE_GPLL	Enables or disables GPLL module in top level.
Rx D-PHY Type	CSI2	Selects the Rx D-PHY interface type. 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 DPHY Clock Gear	RX_GEAR_8	Rx D-PHY clock Gear. The Avant devices only supports Gear 8.
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	Enables Rx D-PHY FIFO Miscellaneous signals. Define this only if Rx FIFO is enabled and Rx FIFO Miscellaneous signals option is checked in the Rx DPHY user interface.
Wait for FS/VSS	WAIT_FS	Waits 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.
	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 the rx_byte_clk cycles. The value must be within the range of 1–8,191.
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 the 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. The Avant devices only support Gear 8.
Tx D-PHY Clock Mode[1]	TX_CLK_MODE_HS_ONLY	Tx D-PHY Clock mode. Only one of these two directives must be defined.
	TX_CLK_MODE_HS_LP	
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.
Sensor Configuration	RESOLUTION {value}	RESOLUTION = 0 (default value), use 1,080p resolution. RESOLUTION = 1, use 4K resolution

Category	Directives	Descriptions
	SENSOR_SLAVE_ADDR {value}	I2C address of Camera module. The default value is 26.
	I2C_HIGH_CYCLE {value}	I2C configuration of camera module. The default value is 35.
	I2C_LOW_CYCLE {value}	I2C configuration of camera module. The default value is 35.
	I2C_GAP_CYCLE {value}	I2C configuration of camera module. The default value is 200.

Notes:

1. The HS_LP mode means non-continuous clock mode and HS_ONLY means continuous clock mode.
2. The HS_LP mode works only if the Rx byte clock for corresponding Rx channel can be generated internally or from an external source.

4.2. Simulation Directives

Table 4.2 shows the simulation directives for this reference design.

Table 4.2. Simulation Directives

Category	Directives	Descriptions
GPLL Lock	GPLL_LOCK_SIM	Enables gpll_lock in the simulation testbench.
DSI Sync Mode	DSI_SYNC_MODE_EVENT	Selects 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	Inserts End of Transmission Packet (EoTp) 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 {value}	Rx DPHY byte clock in ps.
Tx DPHY Byte Clock Period	TX_FREQ_TGT {value}	Tx DPHY byte clock in ps.
Tx DPHY Clock Period	TX_DPHY_CLK_PERIOD {value}	Tx Channel DPHY clock period in ps.
Rx Virtual Channel	RX_VC {value}	Virtual channel ID of the packet. The 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_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	

Category	Directives	Descriptions
	RAW12	
	RGB888	
Pixel per Clock	RX_PEL_PER_CLK {value}	Sets 1, 2 or 4 Pixel per Clock.
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.

5. Signal Description

Table 5.1 shows the top-level I/O of this reference design. Actual I/O depends on your channel and lane configurations. Compiler directives automatically declare all necessary I/O ports.

Table 5.1. CSI-2/DSI Duplicator Top-Level I/O

Port Name	I/O	Width	Description
clk27_i	In	1	27 MHz input clock from the Avant Versa board.
cam_clk_o	Out	1	Output clock to camera sensor.
cam_scl_io	Inout	1	Bidirectional I2C connection for camera sensor configuration.
cam_sda_io	Inout	1	Bidirectional I2C connection for camera sensor configuration.
cam_rst_n_o	Out	1	Output active low reset signal to camera sensor.
avantE_clk27_o	Out	1	27 MHz clock forwarding to the Avant-E device. Use for hardware testing.
avantE_reset_n_o	out	1	Reset forwarding to the Avant-E device. Use for hardware testing.
ext_clk_i	In	1	External input reference clock. It is used to feed a clock to GPLL and Tx D-PHY PLL. This port is declared only when RX_CLK_MODE_HS_LP is defined in synthesis_directives.v.
rx_clk_p_i	Inout	1	Positive differential Rx D-PHY input clock.
rx_clk_n_i	Inout	1	Negative differential Rx D-PHY input clock.
rx_d_p_i [*NUM_RX_LANE*-1:0][1]	Inout	1	Positive differential Rx D-PHY input data 0–3.
rx_d_n_i [*NUM_RX_LANE*-1:0][1]	Inout	1	Negative differential Rx D-PHY input data 0–3.
tx*N*_clk_p_o	Inout	1	Positive differential Tx D-PHY output clock on the Tx channel N[3].
tx*N*_clk_n_o	Inout	1	Negative differential Tx D-PHY output clock on the Tx channel N[3].
tx*N*_d_p_o [*NUM_TX_LANE*-1:0][2]	Inout	1	Positive differential Tx D-PHY output data 0–3 on the Tx channel N[3].
tx*N*_d_n_o [*NUM_TX_LANE*-1:0][2]	Inout	1	Negative differential Tx D-PHY output data 0–3 on Tx channel N[3].
reset_n_i	In	1	Asynchronous active low system reset.

Notes:

1. *NUM_RX_LANE* represents the number of Rx lanes defined in synthesis_directives.v, where NUM_RX_LANE = 1, 2 or 4.
2. *NUM_TX_LANE* represents the number of Tx lanes defined in synthesis_directives.v, where NUM_TX_LANE = 1, 2 or 4.
3. *N* represents the number of Tx channels, where N = 1, 2, 3 or 4.

6. Running the Reference Design

This section describes how to run the Avant 1-to-N MIPI CSI-2/DSI Duplicator reference design using the Lattice Radiant software. For more details on the Lattice Radiant software, refer to the Lattice Radiant Software User Guide.

6.1. Compiling the Reference Design

This section provides the procedure of compiling and creating your FPGA bitstream file using the Lattice Radiant software. However, the reference design package includes a pre-compiled FPGA bitstream file located in the *<design_folder>/demo_bitfiles* directory for you to run the hardware test directly. The full design compilation is required each time you perform IP generation, or any modifications in the design including modification in the synthesis_directives.v.

To compile and generate bitstream file of your design using the Lattice Radiant software, follow these steps:

1. Open the Lattice Radiant software.
2. At the Start Page, click **Open Project** and from the project database, open the Radiant project file (.rdf) from the *<design_folder>/avant_mipi_1toN_duplicator_rd/mipi_dup_rd_avant* directory.
3. Click **Export Files** to proceed with full design compilation and to generate bitstream file. For a successful design compilation, the generated bitstream file (.bit) will be in the *<design_folder>/avant_mipi_1toN_duplicator_rd/mipi_dup_rd_avant/impl_1* folder.

7. Simulating the Reference Design

The reference design testbench files are provided to run the functional simulation using the QuestaSim simulator. You need to modify the synthesis_directives.v and simulation_directives.v files according to your requirements, ensuring they match the settings made in the IP.

To simulate the design, perform the following steps:

1. Click on **Tools > Simulation Wizard**.
2. Provide a Project Name to create a new spf file and click **Next**.

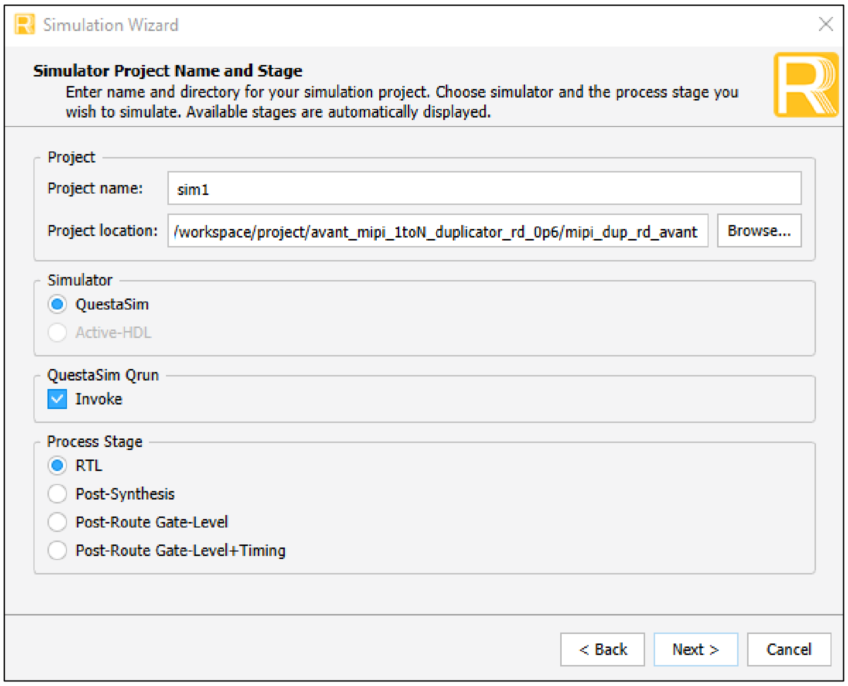

Figure 7.1. Simulation Wizard: Create Simulation Project

3. Add and reorder the simulation source file. Note that for CSI-2 simulation, you need to remove dsi_model.v and dsi_checker_dup.v. Meanwhile for DSI simulation, you need to remove csi2_model.v and csi2_checker_dup.v. Once done, click **Next**.

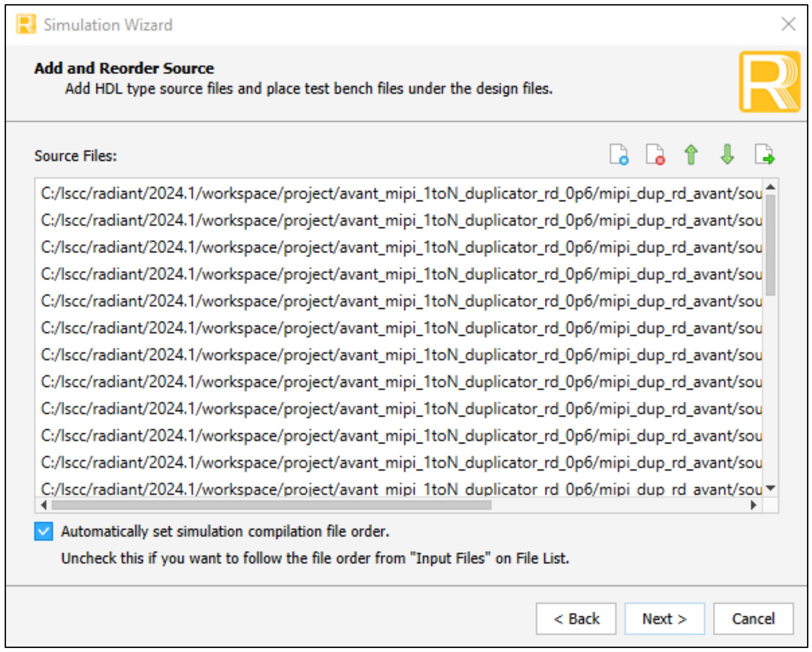

Figure 7.2. Simulation Wizard: Add and Reorder Simulation Source File

4. Set the simulation top module as **top** and click **Next**.

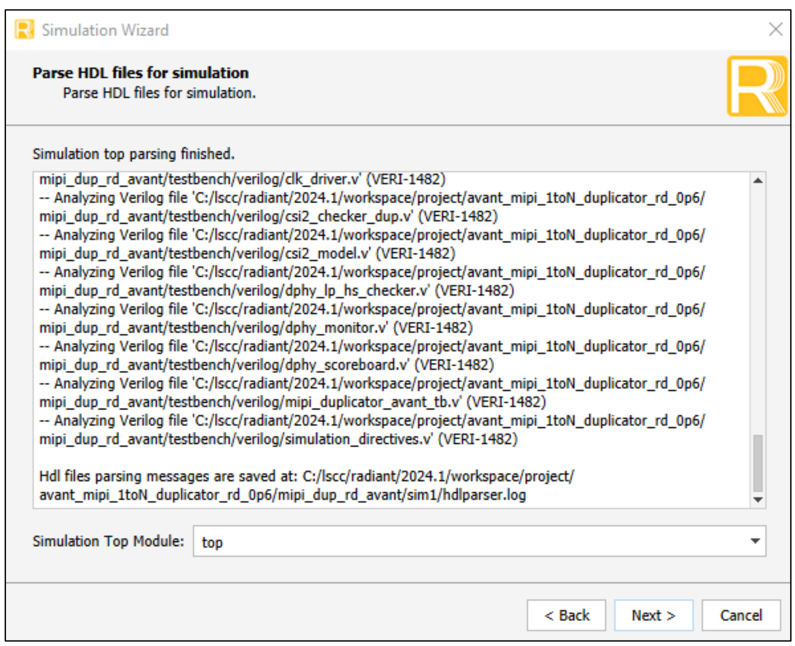

Figure 7.3. Simulation Wizard: Select Simulation Top Module

5. Select the configuration as shown in Figure 7.4. Note that for a complete simulation runtime, the **Default Run** must be set to 0 ns. Click **Finish**.

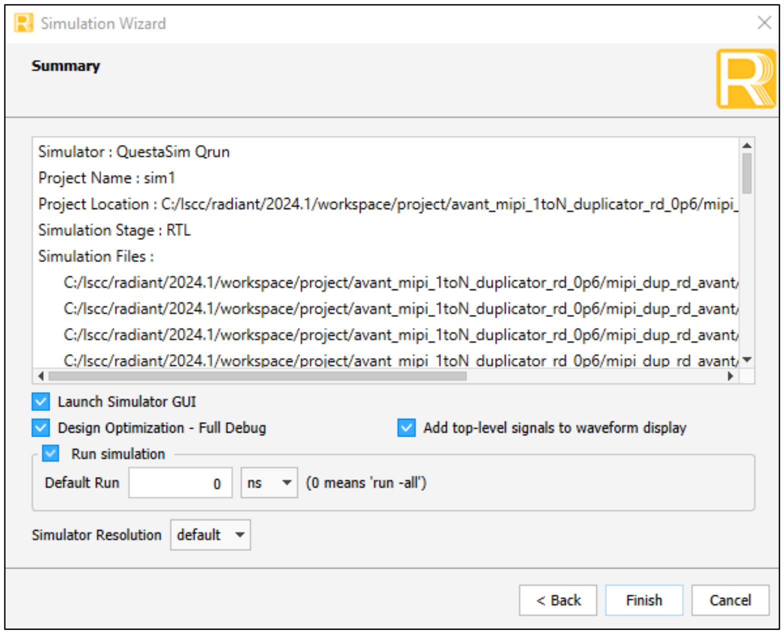

Figure 7.4. Simulation Wizard: Summary Page

6. Wait until the QuestaSim Lattice FPGA Edition software loads. The simulation will immediately start and stop once completed.

7.1. Simulation Results

The reference design simulation testbench takes all the data and compare between the expected data and output data from the reference design, including VC, ECC, and CRC data. It shows the following statements while running and doing data comparison.

```
# [9119267500][DPHY_TX0_CHK] payload data = 24 81 09 63 --- [9119267500][DPHY_TX0_CHK]Data matches RX Data : 24 81 09 63
# [9119267500][DPHY_TX1_CHK] payload data = 24 81 09 63 --- [9119267500][DPHY_TX1_CHK]Data matches RX Data : 24 81 09 63
# [9119267500][DPHY_TX2_CHK] payload data = 24 81 09 63 --- [9119267500][DPHY_TX2_CHK]Data matches RX Data : 24 81 09 63
# [9119267500][DPHY_TX3_CHK] payload data = 24 81 09 63 --- [9119267500][DPHY_TX3_CHK]Data matches RX Data : 24 81 09 63
# [9119933900][DPHY_TX0_CHK] payload data = 0d 8d 65 12 --- [9119933900][DPHY_TX0_CHK]Data matches RX Data : 0d 8d 65 12
# [9119933900][DPHY_TX1_CHK] payload data = 0d 8d 65 12 --- [9119933900][DPHY_TX1_CHK]Data matches RX Data : 0d 8d 65 12
# [9119933900][DPHY_TX2_CHK] payload data = 0d 8d 65 12 --- [9119933900][DPHY_TX2_CHK]Data matches RX Data : 0d 8d 65 12
# [9119933900][DPHY_TX3_CHK] payload data = 0d 8d 65 12 --- [9119933900][DPHY_TX3_CHK]Data matches RX Data : 0d 8d 65 12
# [9120600300][DPHY_TX0_CHK] payload data = 01 0d 76 3d --- [9120600300][DPHY_TX0_CHK]Data matches RX Data : 01 0d 76 3d
# [9120600300][DPHY_TX1_CHK] payload data = 01 0d 76 3d --- [9120600300][DPHY_TX1_CHK]Data matches RX Data : 01 0d 76 3d
# [9120600300][DPHY_TX2_CHK] payload data = 01 0d 76 3d --- [9120600300][DPHY_TX2_CHK]Data matches RX Data : 01 0d 76 3d
# [9120600300][DPHY_TX3_CHK] payload data = 01 0d 76 3d --- [9120600300][DPHY_TX3_CHK]Data matches RX Data : 01 0d 76 3d
# [9121266700][DPHY_TX0_CHK] payload data = ed 8c f9 c6 --- [9121266700][DPHY_TX0_CHK]Data matches RX Data : ed 8c f9 c6
# [9121266700][DPHY_TX1_CHK] payload data = ed 8c f9 c6 --- [9121266700][DPHY_TX1_CHK]Data matches RX Data : ed 8c f9 c6
# [9121266700][DPHY_TX2_CHK] payload data = ed 8c f9 c6 --- [9121266700][DPHY_TX2_CHK]Data matches RX Data : ed 8c f9 c6
# [9121266700][DPHY_TX3_CHK] payload data = ed 8c f9 c6 --- [9121266700][DPHY_TX3_CHK]Data matches RX Data : ed 8c f9 c6
```

Figure 7.5. Example Simulation Result 1

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

```
# [10499381900][DPHY_TX0_CHK] Trail Bytes = ff ff ff ff --- Check OK
# [10499381900][DPHY_TX1_CHK] Trail Bytes = ff ff ff ff --- Check OK
# [10499381900][DPHY_TX2_CHK] Trail Bytes = ff ff ff ff --- Check OK
# [10499381900][DPHY_TX3_CHK] Trail Bytes = ff ff ff ff --- Check OK
# [10509711100][DPHY_TX0_HS_LP_CHK] HS to LP11 Transition on D0 lane with HS-TRAIL period = 109 ns
# [10509711100][DPHY_TX3_HS_LP_CHK] HS to LP11 Transition on D0 lane with HS-TRAIL period = 109 ns
# [10509711100][DPHY_TX2_HS_LP_CHK] HS to LP11 Transition on D0 lane with HS-TRAIL period = 109 ns
# [10509711100][DPHY_TX1_HS_LP_CHK] HS to LP11 Transition on D0 lane with HS-TRAIL period = 109 ns
# [10527703900][DPHY_TX1_HS_LP_CHK] HS to LP11 Transition on clock lane with TRAIL period = 68 ns
# [10527703900][DPHY_TX2_HS_LP_CHK] HS to LP11 Transition on clock lane with TRAIL period = 68 ns
# [10527703900][DPHY_TX3_HS_LP_CHK] HS to LP11 Transition on clock lane with TRAIL period = 68 ns
# [10527703900][DPHY_TX0_HS_LP_CHK] HS to LP11 Transition on clock lane with TRAIL period = 68 ns
#          10680391900 DPHY CSI-2 after frame gap
#
#          10880396100 DPHY CLK CONT : Driving CLK-Trail
#          10886391900 TEST END
#
# TX CH #0 : 6 / 6 HS transmission completed successfully
# TX CH #1 : 6 / 6 HS transmission completed successfully
# TX CH #2 : 6 / 6 HS transmission completed successfully
# TX CH #3 : 6 / 6 HS transmission completed successfully
#
### Simulation Completed ###
```

Figure 7.6. Example Simulation Result 2

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 7.7 shows an example of CSI-2 duplication of four Tx channels with the Tx and Rx channels in non-continuous clock mode. Both Rx and Tx channels have four lanes. Each Rx or Tx lane bandwidth is 1,200 Mb/s, which means total bandwidth of each Rx or Tx channel is 1,200 Mb/s × 4 lane = 4,800 Mb/s. In this case, BD_RD_DLY is set to 10. As shown in the figure below, TX0, TX1, TX2, and TX3 channels have the same behavior and shares the same output.

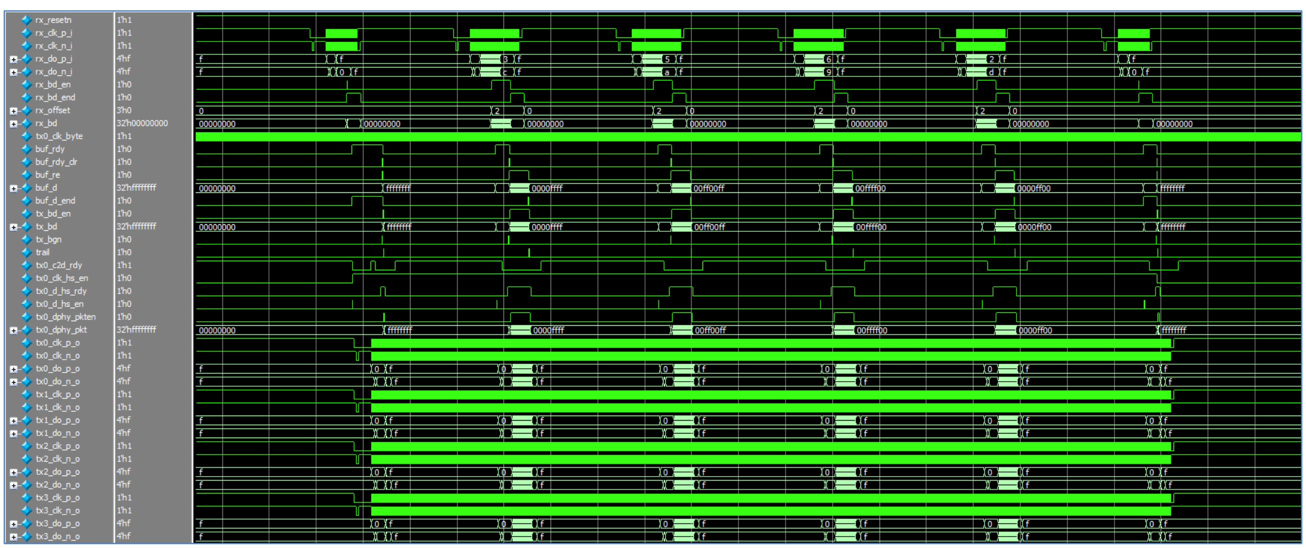

Figure 7.7. Functional Simulation Example of CSI-2 with Four Non-Continuous Clock Mode for Tx Channels

8. Implementing the Reference Design on Board

The Avant 1-to-N CSI-2/DSI Duplicator Reference Design can be evaluated using the following hardware:

- Lattice Avant-X Versa Board
- Lattice Avant-E Evaluation Board Revision D
- Power supply for the Evaluation Board and Versa Board
- HDMI FMC daughter card (Terasic HDMI FMC (Sil9136-3))
- Image sensor (Sony IMX258)
- HDMI sink (Monitor) capable of 1080p60Hz and compatible with CEA-861 timing specifications.
- FMC-to-FMC connector (HDR-169470-01)
- USB cable for device programming

8.1. Hardware Setup

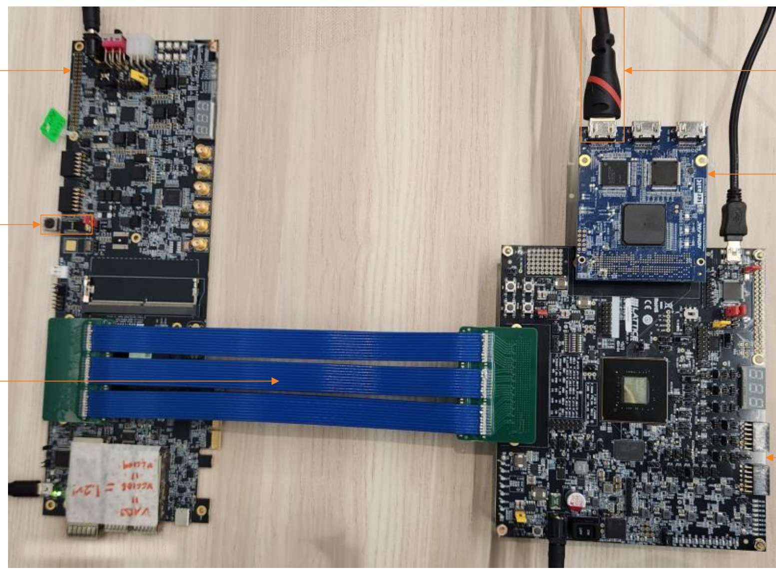

Figure 8.1. Lattice Avant 1-to-N CSI-2/DSI Duplicator Reference Design Hardware Setup

Figure 8.1 shows the full setup required to run the hardware test. The following on-board switch and pin settings must be configured to ensure proper operation:

- Lattice Avant-X Versa Board
  - Change R130 resistor from 1KΩ to 2KΩ to change VCC_VADJ to 1.2V.
  - Connect image sensor to onboard CSI-2 connector (J21).
  - On J22, use 1-2 jumper setting to provide 1.2V power to camera sensor.
  - Connect FMC-to-FMC connector to onboard FMC+ connector (J19).
  - Connect USB cable to onboard Mini USB port for device programming.
  - Provide power to the board via 12V DC power jack.
- Lattice Avant-E Evaluation Board
  - Remove jumper cable JP75 and place it to JP61 to change VCCIO7 from 1.8V to 1.2V.
  - Connect HDMI FMC daughter card to onboard FMC1 connector (J48).
  - Connect HDMI Tx of the HDMI FMC daughter card to a HDMI monitor with HDMI cable.
  - Connect FMC-to-FMC connector to onboard FMC2 connector (J54).
  - Connect USB cable to onboard Mini USB port for device programming.
  - Provide power to the board via 12V DC power jack.

Table 8.1. User Interface on the Avant-X Versa and Avant-E Evaluation Boards

Components	Description
Avant-X Versa Pushbutton SW12	Acts as the main reset for both Avant-X and Avant-E designs on push.
Avant-E Evaluation Pushbutton SW4	Push to toggle the video stream to be displayed. The default channel is Channel 0.
Avant-E Evaluation LED D6	The image sensor I2C configuration is complete.
Avant-E Evaluation LED D7	The HDMI I2C configuration is complete.
Avant-E Evaluation LED D8	The LPDDR4 Memory Controller initialization is complete.
Avant-E Evaluation LED D9	Error detected during LPDDR4 Memory Controller training sequences.
Avant-E Evaluation LED D10	Detected an overlap between lines from the image sensor due to insufficient horizontal blanking.
Avant-E Evaluation LED D11	Detected an overlap between frames from the image sensor due to insufficient vertical blanking.

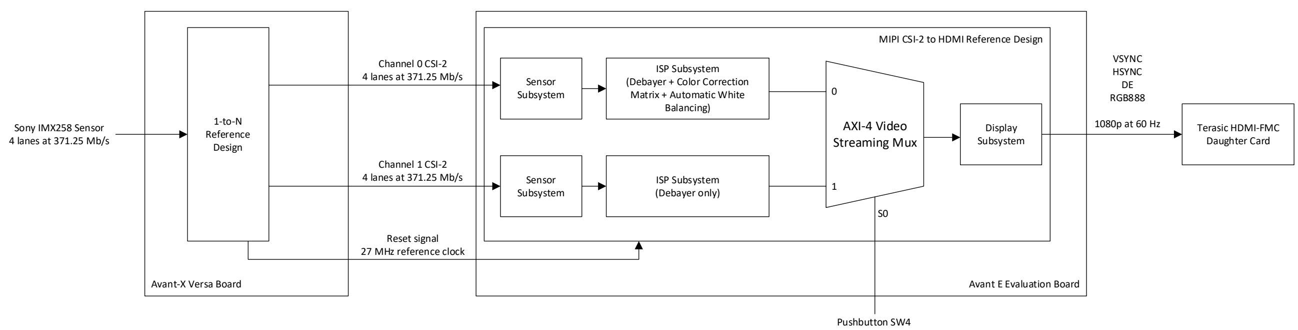

Figure 8.2. Lattice Avant 1-to-N CSI-2/DSI Duplicator Reference Design Hardware Test Block Diagram

Figure 8.2 shows the block diagram of the hardware test setup. Avant-X Versa Board is loaded with the 1-to-N MIPI CSI-2/DSI Duplicator reference design. The design is configured to take four-lanes MIPI D-PHY CSI-2 at 371.25Mb/s bit rate per lane and then being duplicated over to two separate MIPI channels which then would be driving into the Avant-E Evaluation Board.

Meanwhile, the Avant-E Evaluation Board is loaded with a modified version of Avant MIPI CSI-2 to HDMI reference design to take in the data from two channels of MIPI D-PHY CSI-2. One of the channels (Channel 0) goes through the full ISP pipeline, such as Debayer, Color Correction Matrix, and Automatic White Balancing. The other channel (Channel 1) goes through the Debayer only. Both video data streams are merged at a mux, which would be then output to the Display subsystem to generate the necessary video timing to display 1080p video feed at 60 Hz. The pushbutton SW4 is used by the mux to select which video input streams to be sent over. By default, the video stream from Channel 0 path is selected.

8.2. Programming the Board

Hardware evaluation uses two separate reference designs:

- Avant 1-to-N CSI-2/DSI Duplicator Reference Design – this reference design includes a pre-compiled bitstream file (*mipi_dup_1to2_versa.bit*) located in the *<design_folder>/demo_bitfiles* directory for you to run the hardware evaluation. The bitstream file is programmed into the Avant-X Versa Board.
- Custom Avant MIPI-to-HDMI Reference Design – this is the modified version of the Avant MIPI CSI-2 to HDMI Reference Design. To facilitate hardware evaluation, the bitstream file (*mipi2hdmi_2to1_eval.bit*) is provided in the *<design_folder>/demo_bitfiles* of the Avant 1-to-N CSI-2/DSI Duplicator reference design in which it is programmed into the Avant-E Evaluation Board.

Below are the steps to program the Avant-X Versa Board and Avant-E Evaluation Board to perform hardware evaluation:

1. Open the Lattice Radiant software.

2. At the Start Page, click **Open Project** icon and from the project database, open the Radiant project file (.rdf) from the *<design_folder>/avant_mipi_1toN_duplicator_rd/mipi_dup_rd_avant* directory.
3. Open **Radiant Programmer** by clicking the icon.
4. Ensure *Fast Configuration* is selected for the Operation and detected Device matches with the connected FPGA board, that is, *LAV-AT-E70* for Avant-E Evaluation Board and *LAV-AT-X70* for Avant-X Versa Board.
5. Under the **File Name**, select the appropriate bitstream file.

   Note: It is recommended to program the Avant-X Versa Board first before programming the Avant-E Evaluation Board to ensures the upstream device signals such as reset and clock to be available and propagated earlier before the downstream device.
6. Ensure Cable and Port within the Cable Setup section is configured correctly.
7. Click **Program Device** icon to proceed programming the device.

8.3. Hardware Test Result

Figure 8.3. Lattice Avant 1-to-N CSI-2/DSI Duplicator Reference Design Hardware Test Default HDMI Output Display

Figure 8.4. Lattice Avant 1-to-N CSI-2/DSI Duplicator Reference Design Hardware Test HDMI Output Display after Pressing SW4

Figure 8.3 shows the output display after downloading both bitstreams to the boards. The output display is taking the video stream inputs from Channel 0. Figure 8.4 shows the output display after the pushbutton SW4 of the Avant-E Evaluation board is being pressed, which would cause the mux to select Channel 1 video stream input. Note that there is a slight greenish tint on Figure 8.4 as compared to Figure 8.3. Channel 0 video stream has gone through the full ISP to do color correction and white balancing. Meanwhile, Channel 1 video stream has only gone through the debayer, which only involves converting RAW video streams into RGB format.

9. Customizing the Reference Design

9.1. Design Directives

This reference design includes the synthesis_directives.v and simulation_directive.v files. You can customize the settings in these directives according to your design requirements, ensuring they match the defined IP settings such as Rx D-PHY IP, Tx D-PHY IP, and other module settings.

9.2. Rx:Tx Permutations

Table 9.1 shows the possible Rx:Tx permutations and Figure 9.1 shows the calculated bandwidth in the Excel sheet provided with this reference design. Some permutations are excluded due to bandwidth limitations. The Lattice Avant MIPI D-PHY can support only Gear 8 with the maximum lane bandwidth up to 1,200 Mb/s. The permutations shown in Table 9.1 is based on the assumption that total bandwidth (lane bandwidth × 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 Table 9.1 and Figure 9.1 might not directly apply. These cases are described in the rx_buffer section.

Table 9.1. Rx:Tx Permutations

Number of Rx Lane	Rx Gear	Rx Lane Bandwidth x (Mb/s)[1]	Number of Tx Lane	Tx Gear[2]	Tx Lane Bandwidth y (Mb/s)	Tx Byte Clock/ Rx Byte Clock
1	8	$160 \le x \le 1200$	1	8	$160 \le y \le 1200$	1
		$320 \le x \le 1200$	2		$160 \le y \le 600$	0.5
		$640 \le x \le 1200$	4		$160 \le y \le 300$	0.25
2		$160 \le x \le 1200$	2		$160 \le y \le 1200$	1
		$320 \le x \le 1200$	4		$160 \le y \le 600$	0.5
4		$160 \le x \le 1200$	4		$160 \le y \le 1200$	1

Notes:

1. Currently with CSI-2/DSI Receiver v1.7.0 IP, the maximum link rate for the Avant D-PHY is limited to 1,200 Mb/s without the dynamic clock-to-data calibration feature.
2. The Avant devices support Soft D-PHY only and hence they support only Gear 8.

9.3. 1-to-N MIPI CSI-2/DSI Duplicator Calculator

The reference design includes the *mipi_dup_RD_Avant.xlsx* file, located within the *<design_directory>/docs* folder. This tool helps to calculates the Tx and Rx byte clock, the D-PHY clock, and the permissible number of Tx lanes and Tx line rate based on the given configuration.

Avant 1-to-N MIPI CSI-2/DSI Duplicator Reference Design Parameter Calculator		
D-PHY RX Setting		
Number of RX Lanes	1	
RX Gear	8	
RX Line Rate (per lane)	1000	Mbps
RX DPHY Clock Frequency	**500**	MHz
RX Byte Clock Frequency	**125**	MHz
D-PHY TX Setting		
Number of TX Channels	1	**1 - 2 TX channel: Recommend to place primary & secondary D-PHY TX in the same bank for ECLK sharing.**
Number of TX Lanes	4	
TX Gear	8	
TX Line Rate (per lane)	**250**	Mbps
TX DPHY Clock Frequency	**125**	MHz
TX Byte Clock Frequency	**31.25**	MHz
Legend:		
Require User Input	*Calculated Value*	*Error*

Figure 9.1. Configuration Helper Calculator

10. Resource Utilization

Resource utilization depends on the configuration used. Table 10.1 shows the resource utilization examples under certain configurations targeting the LAV-AT-X70 device. This is just a reference; actual usage varies.

The configurations are as follows:

- Radiant Software Version: Lattice Radiant software version 2024.1
- Device: LAV-AT-X70ES-3LFG1156I
- Interface Type: CSI-2
- Link Rate: 1,200 Mb/s
- Tx/Rx Gear: 8
- D-PHY Clock Mode: Continuous clock mode

Table 10.1. Resource Utilization

Configuration	LUT4	FPU Register	EBR	I/O Buffers
Tx/Rx 1-lane, 1 Tx Channel	1180	820	2.5	16
Tx/Rx 2-lanes, 1 Tx Channel	1459	916	2.5	20
Tx/Rx 4-lanes, 1 Tx Channel	1846	1156	2.5	28
Tx/Rx 1-lane, 2 Tx Channel	1337	965	2.5	20
Tx/Rx 2-lanes, 2 Tx Channel	1623	1117	2.5	26
Tx/Rx 4-lanes, 2 Tx Channel	2056	1469	2.5	38
Tx/Rx 1-lane, 3 Tx Channel	1430	1060	2.5	24
Tx/Rx 2-lanes, 3 Tx Channel	1738	1236	2.5	32
Tx/Rx 4-lanes, 3 Tx Channel	2194	1636	2.5	48
Tx/Rx 1-lane, 4 Tx Channel	1533	1155	2.5	28
Tx/Rx 2-lanes, 4 Tx Channel	1853	1355	2.5	38
Tx/Rx 4-lanes, 4 Tx Channel	2333	1803	2.5	58

11. Debugging

This section provides suggested tools that you can use for debugging your design.

11.1. Debug Tools

11.1.1. Reveal Analyzer

The Reveal Analyzer continuously monitors signals within the FPGA for specific conditions that range from simple to complex conditions. When the trigger condition occurs, the Reveal Analyzer saves signal values preceding, during, and following the event for analysis, including a waveform presentation. The data can be saved in the following format:

- Value change dump file (.vcd) that can be used with tools such as Siemens EDA QuestaSim.
- ASCII tabular format that can be used with tools such as Microsoft Excel®.

Before running the Reveal Analyzer, use the Reveal Inserter to add Reveal modules to your design. In these modules, specify the signals to monitor, define the trigger conditions, and other preferred options. The Reveal Analyzer supports multiple logic analyzer cores using hard/soft JTAG interface. You can have up to 15 modules, typically one for each clock region of interest. When the modules are set up, regenerate the bitstream data file to program the FPGA.

During debug cycles, this tool uses a divide and conquer method to narrow down the problem areas into many small functional blocks to control and monitor the status of each block. Refer to the Reveal User Guide for Radiant Software for details on how to use the Reveal Analyzer.

11.1.2. QuestaSim

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

12. Limitation

The following are the known limitations of this reference design:

- ECLK sharing for the secondary D-PHY (Tx channels 3 and 4) must be on the adjacent bank. For instance, if the primary D-PHY is located at bank 5, ECLK can be shared with the secondary D-PHY at banks 4 and 6.
- Due to the above-mentioned limitation, hardware evaluation using the Avant-X Versa Board can only support up to 2 Tx channels (4 lanes each) on the same bank through the FMC port, as the FMC pin bank locations are non-contiguous, namely banks 0, 6, and 9. However, Tx channels 3 and 4 can still be implemented using a different bank.
- The bitstream file of the Avant 1-to-N CSI-2/DSI Duplicator reference design for hardware evaluation provided in this reference design is limited to a specific configuration to match with the Avant MIPI-to-HDMI reference design.

References

- CSI/DSI D-PHY Tx IP User Guide (FPGA-IPUG-02080)
- CSI/DSI D-PHY Rx IP User Guide (FPGA-IPUG-02081)
- Reveal User Guide for Radiant Software
- Avant-E web page
- Avant-G web page
- Avant-X web page
- Avant-E Evaluation Board web page
- Avant-X Versa Board web page
- 1 to N MIPI CSI-2/DSI Duplicator Reference Design web page
- MIPI CSI-2 to HDMI Reference Design web page
- Lattice mVision Solution Stack web page
- Lattice Radiant Software web page
- Lattice Solutions Reference Designs web page
- Lattice Insights web page for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

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

Revision History

Revision 1.0, December 2024

Section	Change Summary
All	Initial release.

www.latticesemi.com