CSI-2/DSI D-PHY Receiver Submodule IP

User Guide

FPGA-IPUG-02025-1.5

November 2021

Disclaimers

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

Contents

1. Introduction 6
1.1. Quick Facts 6
1.2. Features 7
1.3. Conventions 7
1.3.1. Nomenclature 7
1.3.2. Data Ordering and Data Types 7
1.3.3. Signal Names 7
2. Functional Description 8
2.1. I/O Port Definitions 8
2.2. Submodule without Packet Parser 10
2.3. Submodule with Packet Parser 12
2.4. Submodule Description 15
2.4.1. D-PHY Common Interface Wrapper 15
2.4.2. Rx Global Operations Controller 19
2.4.3. Capture Controller (Packet Parser) 20
2.5. Reset and Clocking 21
2.5.1. Clock Domains 21
3. Configuration Options 23
4. IP Generation and Evaluation 25
4.1. Licensing the IP 25
4.2. Getting Started 25
4.3. Generating IP in Clarity Designer 26
4.4. Generated IP Directory Structure and Files 28
4.5. Running Functional Simulation 30
4.6. Simulation Strategies 32
4.7. Simulation Environment 33
4.8. Instantiating the IP 34
4.9. Synthesizing and Implementing the IP 34
4.10. Hardware Evaluation 34
4.10.1. Enabling Hardware Evaluation in Diamond 34
4.11. Updating/Regenerating the IP 35
4.11.1. Regenerating an IP in Clarity Designer 35
References 36
Technical Support Assistance 36
Appendix A. Resource Utilization 37
Appendix B. What is Not Supported 43
Revision History 44

Figures

Figure 2.1. MIPI D-PHY Rx IP captured Data 10
Figure 2.2. Fabric Interface Timing Diagram of the MIPI Rx D-PHY IP without Packet Parser 10
Figure 2.3. Block Diagram of MIPI Rx D-PHY IP without Packet Parser 11
Figure 2.4. Rx D-PHY with Packet Parser Output Timing Diagram 12
Figure 2.5. Block Diagram of MIPI Rx D-PHY IP with Packet Parser 13
Figure 2.6. Output Timing Diagram with Valid Second Set of Packet Information 14
Figure 2.7 Rx D-PHY Submodule Components 15
Figure 2.8 MIPIDPHYA Instantiation 16
Figure 2.9 RX_FIFO SINGLE 17
Figure 2.10 RX_FIFO PINGPONG 18
Figure 2.11. RX_FIFO QUEUE 19
Figure 2.12 Aligner Modules 19
Figure 2.13. MIPI D-PHY Clock Lane Module State Diagram 20
Figure 2.14. MIPI D-PHY Data Lane Module State Diagram 20
Figure 2.15. Clock Domain Diagram 21
Figure 4.1. Clarity Designer Window 25
Figure 4.2. Starting Clarity Designer from Diamond Design Environment 26
Figure 4.3. Configuring CSI-2/DSI D-PHY Receiver Submodule IP in Clarity Designer 27
Figure 4.4. Configuration Tab in IP Interface 28
Figure 4.5. CSI-2/DSI D-PHY Receiver Submodule IP Directory Structure 29
Figure 4.6. Simulation Environment Block Diagram 32
Figure 4.7. DSI Model Video Data 33
Figure 4.8. CSI-2 Model Video Data 33
Figure 4.9. Regenerating IP in Clarity Designer 35

Tables

Table 1.1. CSI-2/DSI D-PHY Receiver Submodule IP Quick Facts 6
Table 2.1. MIPI D-PHY Receiver Submodule IP Pin Function Description 8
Table 2.2. Clock Frequency Calculations 21
Table 3.1. Configuration Options 23
Table 4.1. Files Generated by Clarity Designer 29
Table 4.2. Testbench Directives Common for DSI and CSI-2 30
Table 4.3. Testbench Directives for D-PHY Timing Parameters 31
Table 4.4. Testbench Directives for DSI Rx Type 31
Table 4.5. Testbench Directives for CSI-2 Rx Type 32
Table A.1. Resource Utilization[1, 3] 37

1. Introduction

The Lattice Semiconductor CSI-2/DSI D-PHY Receiver Submodule IP converts DSI or CSI-2 data to either 64-bit or 32-bit data for Lattice Semiconductor CrossLink™ and CrossLinkPlus™ devices. This is useful for wearable, tablet, human-machine interfacing, medical equipment, and many other applications.

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

The CSI-2/DSI D-PHY Receiver Submodule IP is intended for use in applications that require a D-PHY receiver in the FPGA logic.

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

1.1. Quick Facts

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

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

		MIPI D-PHY Receiver Submodule IP Configuration			
		Gear 16, Soft D-PHY	Gear 8, Soft D-PHY	Gear 16, Hard D-PHY	Gear 8, Hard D-PHY
Core Requirements	FPGA Families Supported	CrossLink/CrossLinkPlus			
Resource Utilization	Targeted Device	LIF-MD6000-6MG81I			
	Data Path Width	64 bits	32 bits	64 bits	32 bits
	LUTs	1566 (CSI-2) 1772 (DSI)	1109 (CSI-2) 1136 (DSI)	2023 (CSI-2) 2254 (DSI)	1245 (CSI-2) 1275 (DSI)
	sysMEM™ EBRs	8	6	8	6
	Registers	503 (CSI-2) 550 (DSI)	450 (CSI-2) 456 (DSI)	708 (CSI-2) 760 (DSI)	554 (CSI-2) 526 (DSI)
	Programmable I/O	10	10	0	0
	MIPI D-PHY	0	0	1	1
Design Tool Support	Lattice Implementation	Lattice Diamond® 3.11 SP1			
	Synthesis	Lattice Synthesis Engine			
		Synopsys® Synplify Pro® N-2018.03L-SP1-1			
	Simulation	Aldec® Active-HDL™ 10.5 Lattice Edition			

Note: The distributed RAM utilization is accounted for in the total LUT4 utilization. The actual LUT4 utilization is distribution among logic, distributed RAM, and ripple logic.

Configurations in Table 1.1 are all 4-lane, with aligner modules and packet parser enabled and Single Mode RX_FIFO.

1.2. Features

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

- Compliant with MIPI D-PHY v1.1, MIPI DSI v1.1, and MIPI CSI-2 v1.1 Specifications
- Selection between Hard Rx D-PHY or Soft Rx D-PHY implementation
- Supports MIPI D-PHY interface from 80 Mb/s up to 1.5 Gb/s (Hardened D-PHY / Gear 16)
- Supports 1, 2, 3, or 4 data lanes and one clock lane
- Supports continuous and non-continuous MIPI D-PHY clock
- Supports all MIPI DSI video mode of operation
- Non-burst mode with Sync pulses
- Non-burst mode with Sync events
- Burst mode
- Optional packet parsing or parallel data translation only
- Supports all MIPI DSI compatible video formats
- Supports all MIPI CSI-2 compatible video formats

1.3. Conventions

1.3.1. Nomenclature

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

1.3.2. Data Ordering and Data Types

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

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

1.3.3. Signal Names

Signal names that end with:

- _*n* are active low
- _*i* are input signals
- Some signals are declared as bidirectional (I/O) but are only used as input. Hence, _*i* identifier is used.
- _*o* are output signals
- Some signals are declared as bidirectional (I/O) but are only used as output. Hence, _*o* identifier is used.
- _*io* are bidirectional signals

2. Functional Description

The MIPI D-PHY Interface for CSI-2/DSI Submodule IP can be configured to include a packet parser, which decodes the contents of the incoming CSI-2 or DSI packets, or to simply output the D-PHY packets without decoding its contents.

Table 2.1 describes the functionality of the IP block that includes the packet parser.

2.1. I/O Port Definitions

Table 2.1. MIPI D-PHY Receiver Submodule IP Pin Function Description

Port Name	Direction	Function Description
Clock and Reset		
reset_n_i	I	Asynchronous system reset (active low)
clk_byte_fr_i	I	Continuously running byte clock. This should be div8 (in gear16) or div4 (in gear8) of the input D-PHY clock. This also clocks the logic that detects the Rx D-PHY data lane transitions (lp_hs_ctrl_d0-3 modules). This is also used by the word_align, lane_align and capture_control modules. Payload output is also in this clock domain.
reset_byte_fr_n_i	I	Active low reset. Resets the nets in the clk_byte_fr clock domain. The signal driving this port must already be synchronized to the clk_byte_fr.
clk_lp_ctrl_i	I	Clocks the logic that detects the Low Power states of the D-PHY clock lane. The period of this clock should be smaller than the tLPX, with enough setup and hold time, to properly sample the Low Power state transitions. No need to drive this reset if the Rx clock mode is HS_ONLY.
reset_lp_n_i	I	Active low reset. This resets the nets in the clk_lp_hs_ctrl clock domain. The signal driving this port must already be synchronized to the clk_lp_hs_ctrl. No need to drive this reset if the Rx clock mode is HS_ONLY.
clk_byte_o	O	Generated byte clock from the D-PHY module based on the input D-PHY clock lane, used to latch the internal parallel byte data from dphy_rx_wrap. This is div4 or div8 of the D-PHY clock lane frequency. This is only active when the data lanes are in high-speed mode.
reset_byte_n_i	I	Active low reset. Resets the logic clocked by the clk_byte_o.
clk_byte_hs_o	O	Generated byte clock from the D-PHY module based on the input D-PHY clock lane, active only when the clock lanes are in high-speed mode. This clock is the same as the clk_byte_o when the submodule is in HS_ONLY mode and D-PHY implementation is Soft DPHY. This may be connected to the clk_byte_fr_i when the RX Clock Mode is HS_ONLY.
pd_dphy_i	I	Active high. Power Down control signal. Applicable only for Hard D-PHY implementation.
MIPI Interface		
clk_p_i, clk_n_i	I/O	MIPI D-PHY clock lane
d0_p_io, d0_n_io	I/O	MIPI D-PHY data lane 0. Available only for MIPI DSI configuration.
d0_p_i, d0_n_i	I/O	MIPI D-PHY data lane 0. Available only for MIPI CSI-2 configuration.
d1_p_i, d1_n_i	I/O	MIPI D-PHY data lane 1. Available only for configurations with two or more data lanes.
d2_p_i, d2_n_i	I/O	MIPI D-PHY data lane 2. Available only for configurations with four data lanes
d3_p_i, d3_n_i	I/O	MIPI D-PHY data lane 3. Available only for configurations with four data lanes.
Fabric Interface (For Low-Power Communications)		
lp_d0_rx_p_o, lp_d0_rx_n_o	O	Low-power values of data lane 0 true and differential lines, respectively.
lp_d1_rx_p_o, lp_d1_rx_n_o	O	Low-power values of data lane 1 true and differential lines, respectively.
lp_d2_rx_p_o, lp_d2_rx_n_o	O	Low-power values of data lane 2 true and differential lines, respectively.

Port Name	Direction	Function Description
lp_d3_rx_p_o, lp_d3_rx_n_o	O	Low-power values of data lane 3 true and differential lines, respectively.
lp_d0_tx_p_i, lp_d0_tx_n_i	I	Input port to drive the low-power values of data lane 0 when low-power data transmission is enabled (for reverse direction communications). Available in DSI Rx Interface only.
lp_d0_tx_en_i	I	When driven high, this puts the data lane 0 in transmit mode for LPDT reverse direction communications. This must be driven low during forward direction. Available in DSI Rx Interface only.
Fabric Interface (Without Packet Parser)		
bd0_o	O	Byte Data directly from lane 0. This is 8-bit wide for gear 8, 16 bits for gear 16.
bd1_o	O	Byte Data directly from lane 1. This is 8-bit wide for gear 8, 16 bits for gear 16.
bd2_o	O	Byte Data directly from lane 2. This is 8-bit wide for gear 8, 16 bits for gear 16.
bd3_o	O	Byte Data directly from lane 3. This is 8-bit wide for gear 8, 16 bits for gear 16.
capture_en_o	O	Indicates valid byte data in bd0/1/2/3_o.
Fabric Interface (With Packet Parser)		
sp_en_o	O	Active high. Short packet detect.
lp_en_o	O	Active high. Long packet detect.
lp_av_en_o	O	lp_av_en_o asserts only when the captured data type matches ref_dt_i. Active high.
vc_o	O	Virtual Channel Identifier. This corresponds to the header word0[7:6] of the received packet.
wc_o	O	Word Count field. This corresponds to the concatenated bits of header word1 and word2 of the received packet.
dt_o	O	Data Type field. This corresponds to the header word0[5:0] of the received packet.
ecc_o	O	Error Correction Code. This corresponds to the header word3[7:0] of the received packet.
payload_o	O	Payload data
payload_en_o	O	Indicates valid payload data in payload_o. Active high.
ref_dt_i	I	Reference data type
Miscellaneous Signals (Status and Debug)		
pll_lock_i	I	PLL lock indicator, if a PLL is used to generate a free-running byte clock. Set this to 1 if a PLL is not used. Active high.
bd_o	O	Valid only for Rx gear 8. This is the DPHY byte data.
term_clk_en_o	O	Active-high enable signal for the line termination of the D-PHY clock lane. This is asserted on detection of transition from LP-11 to LP-01 of the clock lane, and deasserts upon detection of LP-11 after a high-speed mode.
hs_d_en_o	O	Active-high high-speed mode enable for data lane d0. For Hard D-PHY IP, this signal is also used for HS mode enable for the other data lanes.
cd_d0_o	O	Contention detection indicator. Active high.
hs_sync_o	O	This indicates the successful detection of the synchronization code ‘B8 in the data lanes. This signal asserts from the start of synchronization pattern ‘B8 up to the last data captured before detecting LP-11 state (of any lane, if Soft D-PHY; of data lane0 for Hard D-PHY). Active high.
lp_hs_state_clk_o	O	2-bit state encoding of the D-PHY clock controller
lp_hs_state_d_o	O	2-bit state encoding of the D-PHY data lane 0 controller

There is another set of fabric interface signals when the packet parser is enabled and the payload_o data width is greater than 32 bits. This set is valid when there are two packets received within 1-byte clock cycle. This is discussed in detail in subsequent sections.

2.2. Submodule without Packet Parser

This submodule can be instantiated without the packet parser. In this configuration, the output is the received bytes from the D-PHY starting from the reception of the Start of Transmit (SoT) code in all the active data lanes until the detection of the LP-11 signifying the end of high-speed transmission. This includes the trail bits and any glitches that may be seen on the D-PHY bus. It is up to the interfacing logic to obtain and decode the valid data packets from the trail. This configuration is useful for bridging D-PHY packets without going to the protocol level.

In Figure 2.1, the data reflected to the output side is highlighted in gray.

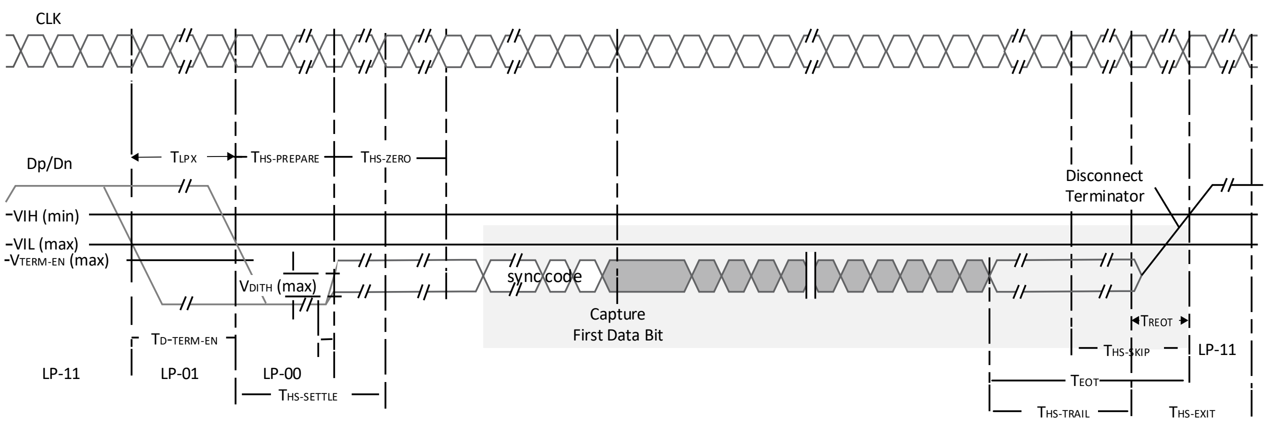


Figure 2.1. MIPI D-PHY Rx IP captured Data

Figure 2.2 shows the fabric interface timing diagram.

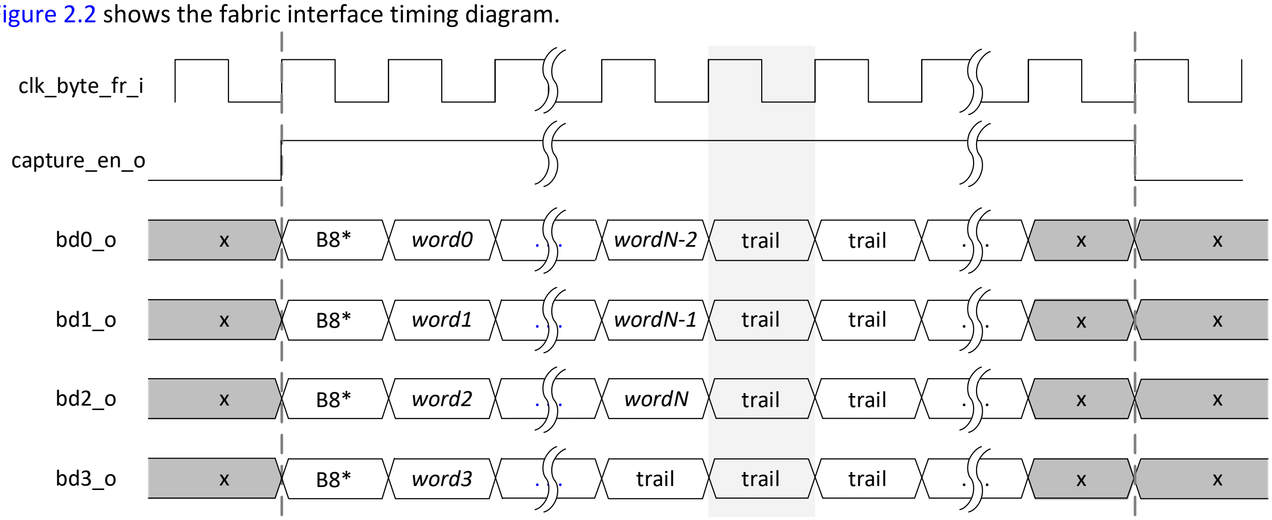


Figure 2.2. Fabric Interface Timing Diagram of the MIPI Rx D-PHY IP without Packet Parser

There must be at least one clock cycle where all the data lanes are sending trail data (shown in italics in Figure 2.2). HS-skip logic is also not provided in this submodule. It is assumed that the parts of data that the interfacing logic decodes are valid by parsing the data type and word count fields.

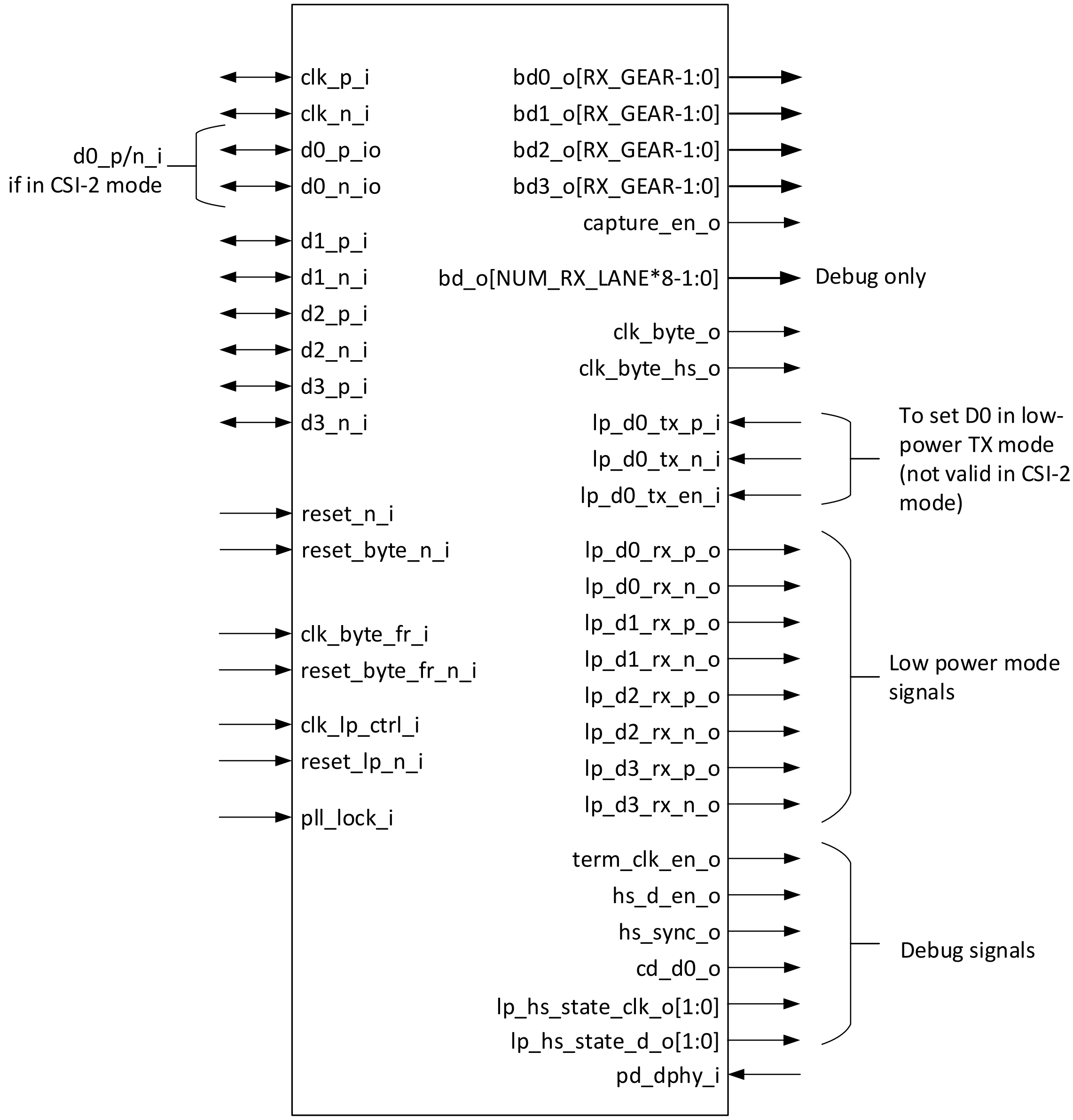


Figure 2.3. Block Diagram of MIPI Rx D-PHY IP without Packet Parser

2.3. Submodule with Packet Parser

This submodule can be instantiated with a DSI or a CSI-2 packet parser. The parser checks the incoming data for a valid data type and corresponding packet fields.

Figure 2.4 shows the timing diagram of the submodule interface with the packet parser enabled.

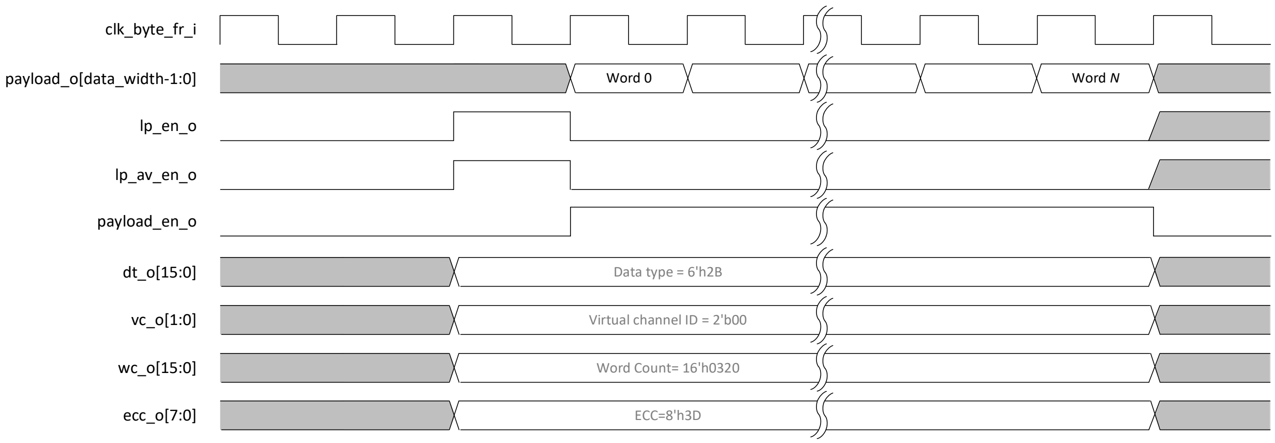


Figure 2.4. Rx D-PHY with Packet Parser Output Timing Diagram

The lp_en_o or sp_en_*o* signal asserts when a valid data type is received. These signals also indicate the valid data type, virtual channel ID, word count, and ECC fields.

The lp_av_en_o only asserts with the lp_en_o, if the long packet received is the same as the input reference data type ref_dt_i. This is to differentiate active video packets from other long packets, such as null or blanking. Consequently, this signal does not assert on any video data type other than the defined ref_dt_i value.

The payload_en_o signal indicates that the data in the payload_o bus contain the valid payload bytes. The width of the payload, data_width, is the number of gear bits multiplied by the number of data lanes. Upper data bytes for the last payload data must be ignored if the word count is not a multiple of data_width/8. The interfacing module should be responsible for extracting the correct payload bytes based on the valid output word count *wc_o*.

Figure 2.5 shows the block diagram of the submodule with the packet parser included.

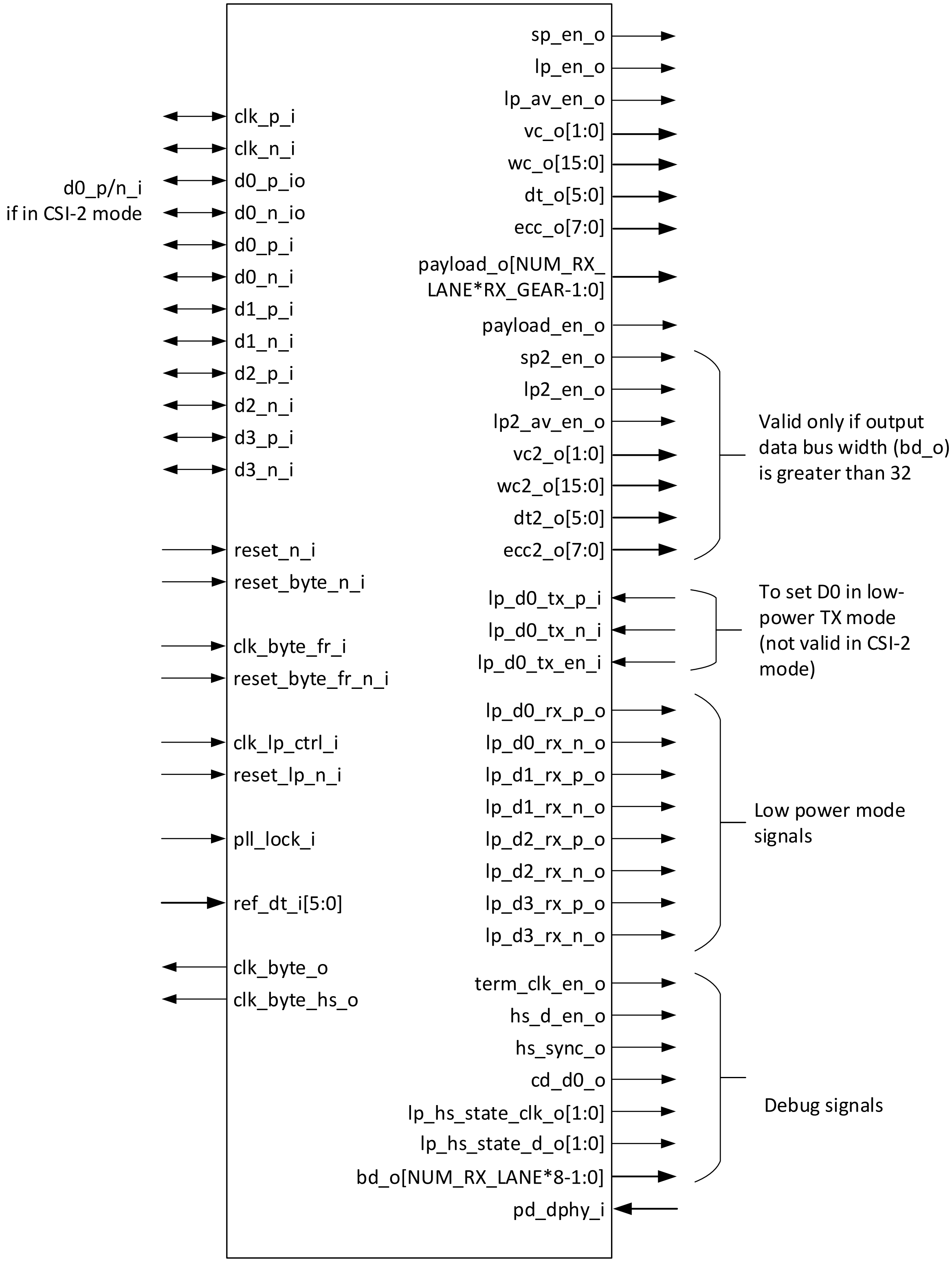


Figure 2.5. Block Diagram of MIPI Rx D-PHY IP with Packet Parser

As shown in the block diagram, there is a second set of decoded packet fields. This set is valid for DSI when the input data bus going to the packet parser is greater than 32 (4-lane with gear 16). In this configuration, two packet headers may simultaneously be decoded within the same byte clock cycle. This scenario is illustrated in Figure 2.6.

The interval between the assertion of the sp_en_o or the lp_en_o and the reception of the valid input data are not fixed; this depends on the gearing and number of lanes.

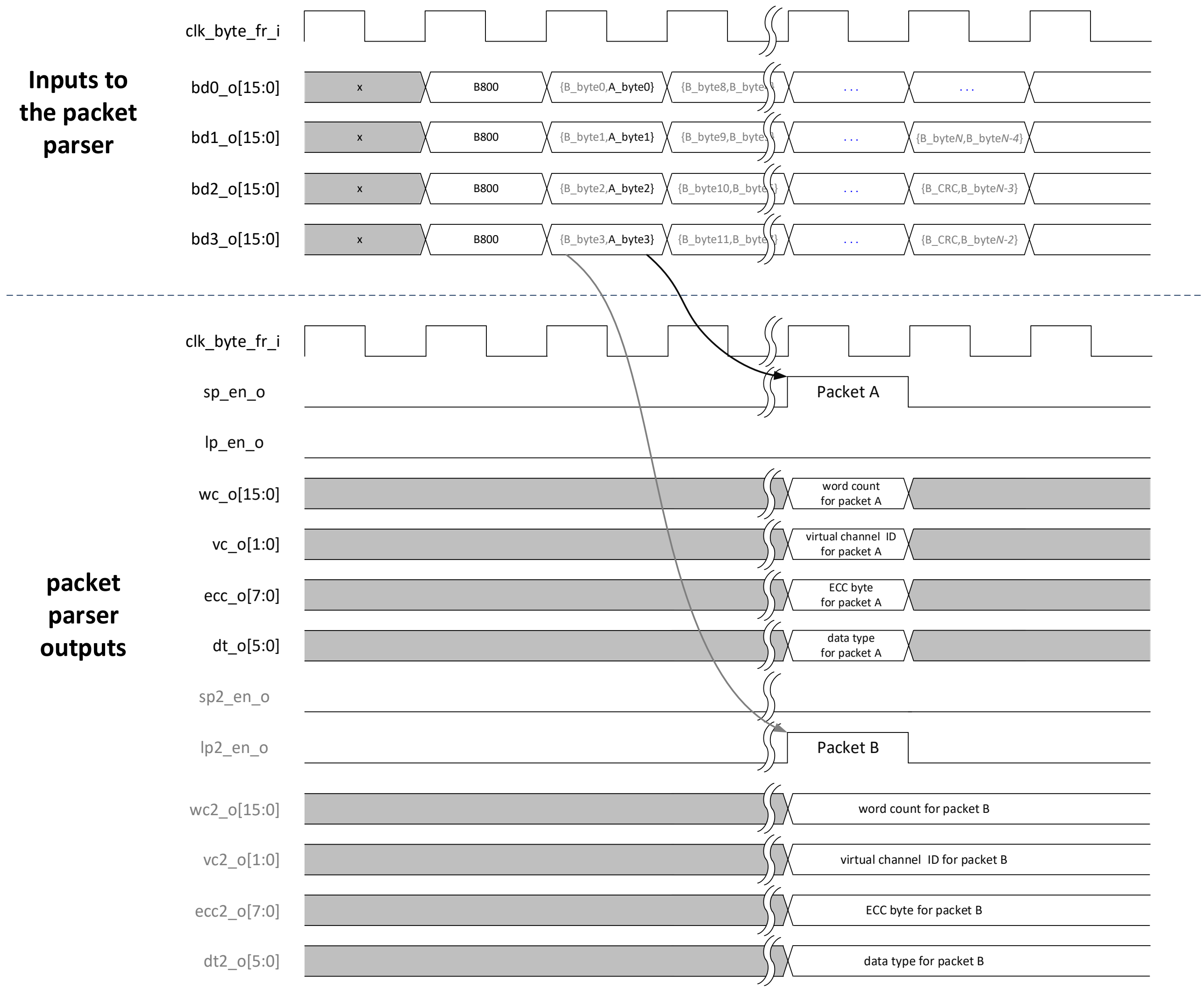


Figure 2.6. Output Timing Diagram with Valid Second Set of Packet Information

2.4. Submodule Description

Figure 2.7 shows the three major blocks of the Rx D-PHY submodule.

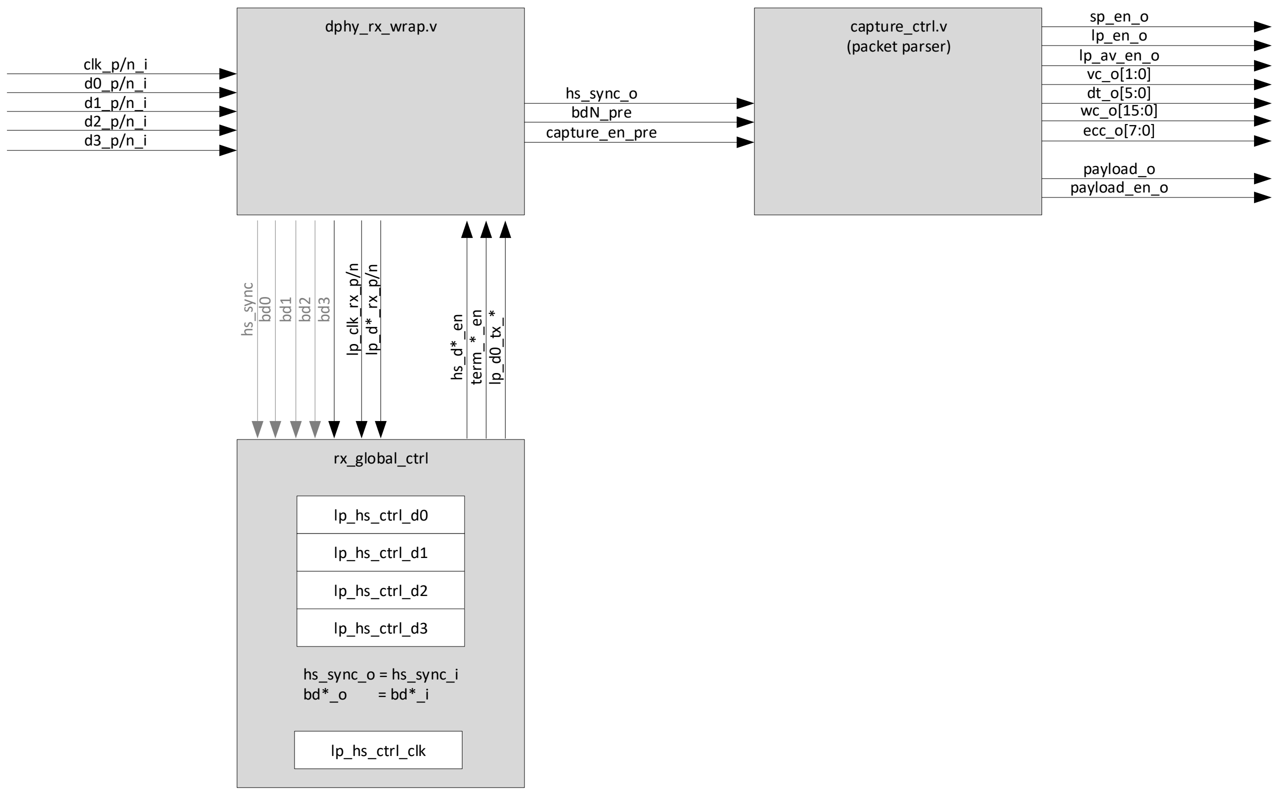


Figure 2.7 Rx D-PHY Submodule Components

2.4.1. D-PHY Common Interface Wrapper

The D-PHY Common Interface Wrapper instantiates and configures either the hard or the soft D-PHY IP (DPHY Module) to receive MIPI D-PHY high-speed data from all enabled data lanes. The interface to the external logic remains the same regardless of the implementation. This block is labeled as dphy_rx_wrap module.

2.4.1.1. MIPIDPHYA

The MIPIDPHYA is the hardened D-PHY block instantiated when the RX DPHY IP user interface parameter is configured as Hard D-PHY. It has an output byteclock, clk_hs_o, derived from the geared down D-PHY input clock lane. This clock is continuously running as long as the input clock lane is continuously running in high speed mode (DPHY Clock Mode = Continuous). This output clock signal may be used as the source of the freerunning input byteclock (clk_byte_fr_i) for other fabric logic. Another clock signal, clk_byte_o, is generated by the hardened block to strobe its output data bytes.

When the Word Aligner module is disabled, the IP uses the aligner within the hardened block MIPIDPHYA. The output of the hardened block is already aligned to the Start-of-Transmit pattern (SoTp), with the SoTp bits already stripped off. To mimic the behavior of the Soft D-PHY, there is a logic within the dphy_rx_wrap that appends the SoTp in the output data bytes. The strobe clk_byte_o is only active after the hard block has detected the SoTp in the data lane0, and goes inactive when the D-PHY data lane0 goes to LP-11 state. Due to the dependency of the clk_byte_o to the SoTp detection, the phase relationship between the two clocks clk_hs_o and clk_byte_o is not constant.

When the Word Aligner module is enabled, the aligner within the MIPIDPHYA is bypassed and the IP uses the word aligner module instead. The parallel data output of the hardened block still contains the zero padding and the unaligned SoTp bits. The strobe clk_byte_o is active once the high speed enable signal is received and goes inactive when the D-PHY data lanes go to LP-11 state.

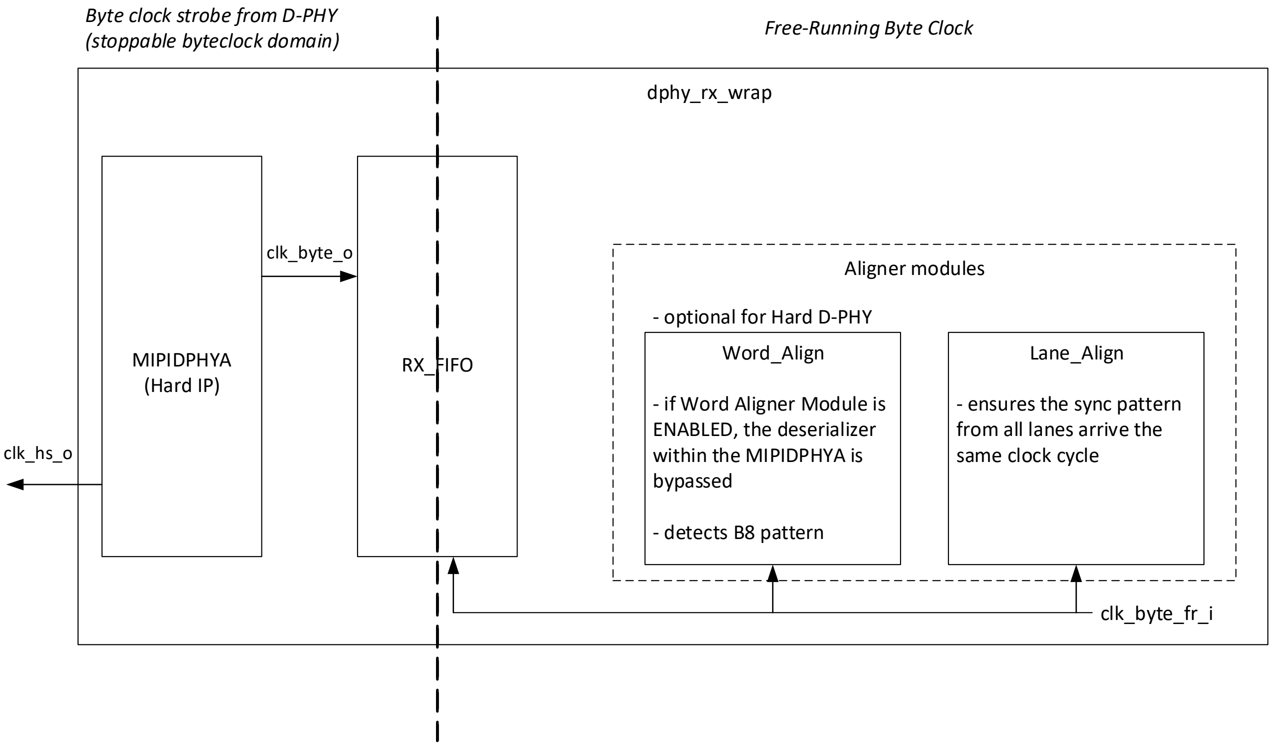


Figure 2.8 MIPIDPHYA Instantiation

2.4.1.2. RX_FIFO

The RX_FIFO is used to transfer parallel data from all lanes from the stoppable byte clock domain (clk_byte_o) to the continuous byte clock domain.

In the case of the Soft DPHY implementation, the RX_FIFO is instantiated between the PHY blocks and the Word Aligner Module. Data being buffered includes the hs-zero bytes before the SoTp, the actual packets, the trail bits and sometimes also the data right before the D-PHY lanes transition to LP-11.

The various implementation types are discussed below.

2.4.1.3. RX_FIFO OFF

If the clk_byte_fr_i has the same frequency and is synchronous with the clock strobe clk_byte_o, then the RX_FIFO can be removed. In the actual generated design, there is still a fixed 4-deep single clock FIFO implemented as LUTs.

This is the recommended setting when the DPHY module is using Soft D-PHY implementation with the D-PHY clock running continuously in high speed mode and the clk_byte_fr_i is driven by the clk_byte_hs_o.

2.4.1.4. RX_FIFO_TYPE = SINGLE

This FIFO acts as an elastic buffer for the data coming from the DPHY Module. The depth is parameterized and configurable through the user interface. This type is recommended for DSI with high speed blanking.

The time it takes to read the data from the FIFO is also configurable. If the packet delay = 0, the whole high speed data bytes, including Null and Blanking packets, are buffered before reading out from the FIFO. The depth must be large enough to accommodate all the data received within the high speed transfer.

If packet delay is non-zero, this enables an internal counter in the clk_byte_fr_i domain that would start counting when the FIFO deasserts the empty flag. Because the MIPI D-PHY protocol itself does not allow data throttling when high speed transfer is already on-going, the depth of the FIFO and the packet delay must be configured properly to ensure the FIFO does not overflow or underflow.

In both cases, the read from FIFO would continue until the empty flag asserts. The interval between high speed transactions (low-power blanking) must be long enough to ensure the FIFO is already empty before the next one is written, otherwise, the FIFO assumes it is still part of the previous data stream. This causes the word aligner to miss the SoTp of the second data stream and interpret the packets erroneously.

Figure 2.9 illustrates the contents of the FIFO when a high-speed blanking DSI stream is being buffered using the RX_FIFO SINGLE.

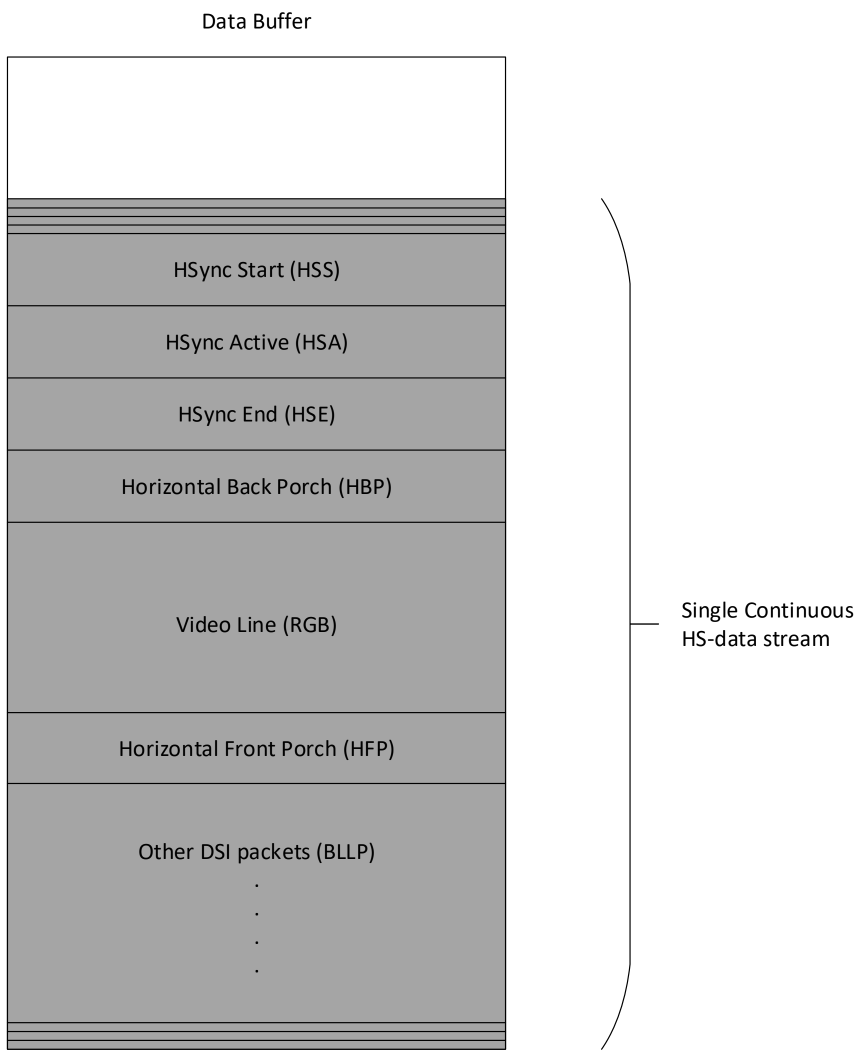


Figure 2.9 RX_FIFO SINGLE

2.4.1.5. RX_FIFO_TYPE = PINGPONG

This type instantiates two (2) dual_clock FIFOs that alternately stores data every high speed transaction. Each data buffers must be good enough to hold the largest data within a high speed transaction, including the hs-zero, SoTp and trail bits.

Similar with the Single type, this also has a parameterized delay before reading out from the buffer to maintain intervals between packets. If packet delay = 0, read starts once its empty signal deasserts and the other one is not busy with read. If packet delay is non-zero, read from that buffer starts once the delay value is met and the other one is not busy with read. This implementation does not track the number of entries within the buffers; read stops once it sees the empty flag. Each buffer is reset after its read operation.

This type is more suitable for high speed transfers with short intervals because data is written alternately between the two buffers. This is recommended for DSI with low power blanking.

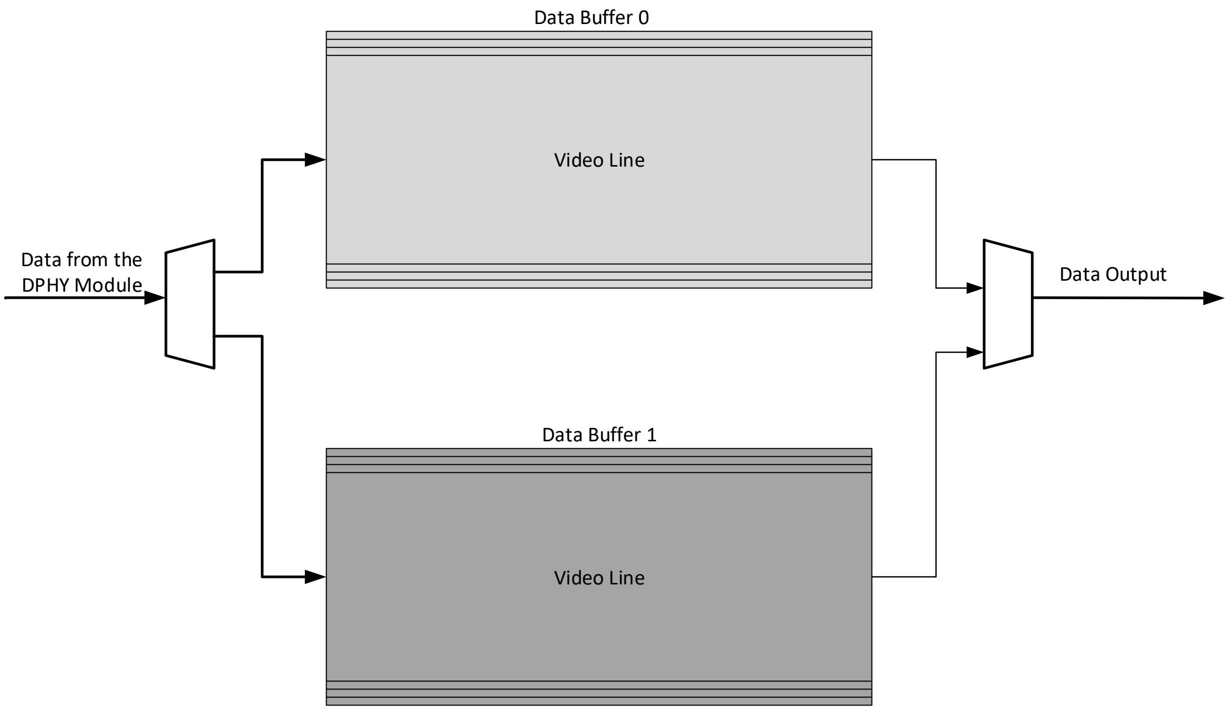


Figure 2.10 RX_FIFO PINGPONG

2.4.1.6. RX_FIFO_TYPE = Queue

This type instantiates one dual-clock FIFO. This also acts as a circular buffer that must be able to hold data from multiple high speed transactions.

Unlike the other two types, this does not have a delay counter. Instead, HS data is buffered completely and a counter tracks the number of rows written during the high speed transaction. This count is stored in an entry queue.

When there is a valid entry in the entry queue, read from the data buffer is triggered. The number of read cycles from the data buffer corresponds to the entry read from the entry queue. This enables the FIFO controller to distinguish the boundaries between successive HS transactions. This introduces significant latency on the first video line, but this also enables the IP to support short intervals between HS transactions.

This is suitable for CSI-2, wherein the packet intervals are not critical, but the intervals between successive high speed transactions are short.

Figure 2.11 illustrates a sample entry within a 4-deep entry queue for a CSI-2 sequence.

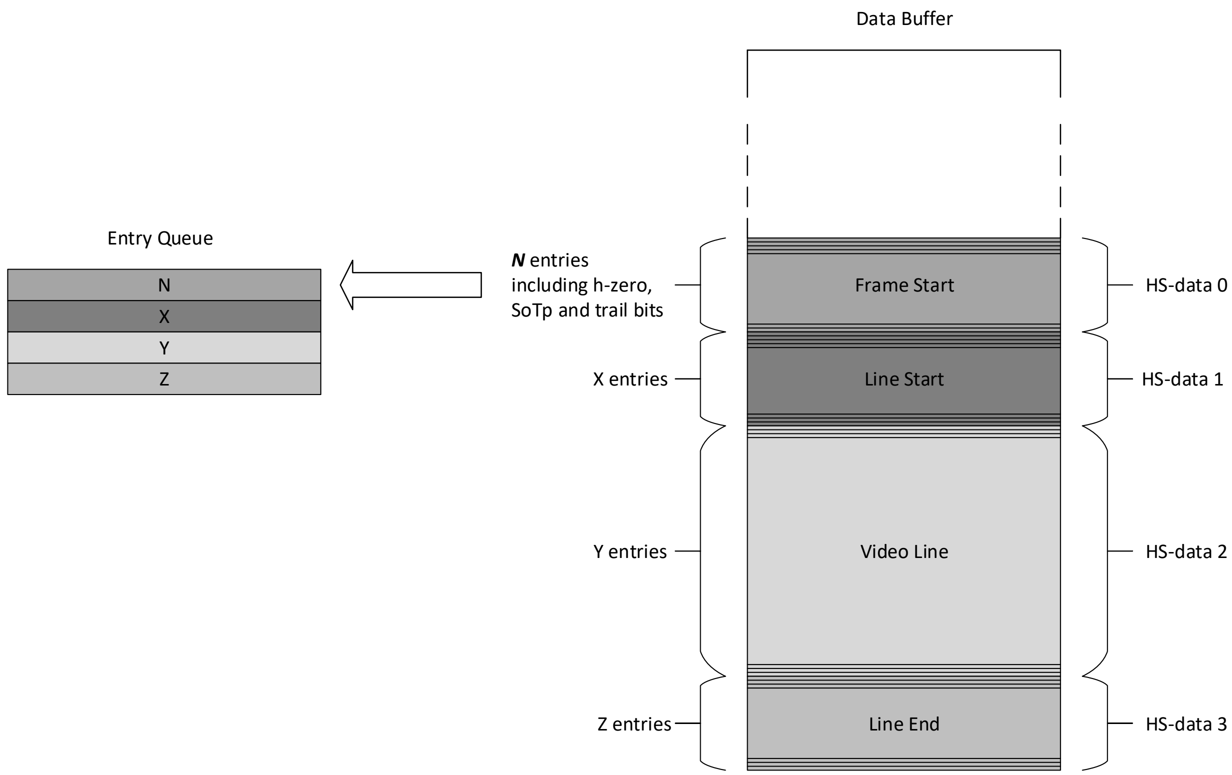


Figure 2.11. RX_FIFO QUEUE

2.4.1.7. WORD_ALIGN and LANE_ALIGNER MODULES

A word aligner logic detects the SoT pattern from each lane and ensures the parallel data are word (byte) aligned. The design assumes that input data lanes are driven at the same time, and skew between data lanes are less than 1 UI. However, due to clock domain crossing or parallel data across the data lanes, a lane aligner module is added to further ensure that the lanes are aligned with each other.

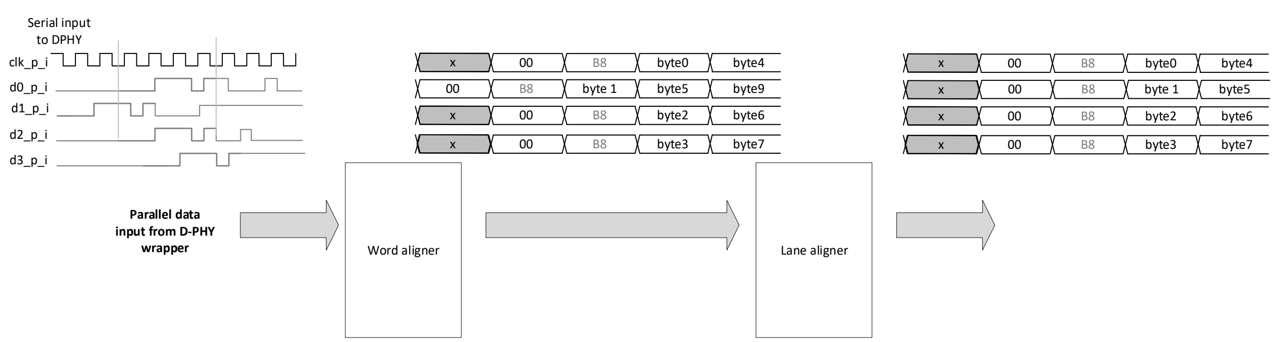


Figure 2.12 Aligner Modules

2.4.2. Rx Global Operations Controller

This block controls the high-speed termination enables of MIPI D-PHY clock and data lanes. When MIPI D-PHY clock is continuous, the High Speed (HS) termination enable of clock lane is tied to VCC. When MIPI D-PHY clock is non-continuous, the HS termination enable of clock lane becomes active right after proper Low Power (LP) to HS transition is observed. A reference clock input is required for this function. Figure 2.13 shows the required LP to HS transition on clock lane as per MIPI D-PHY Specification version 1.1.

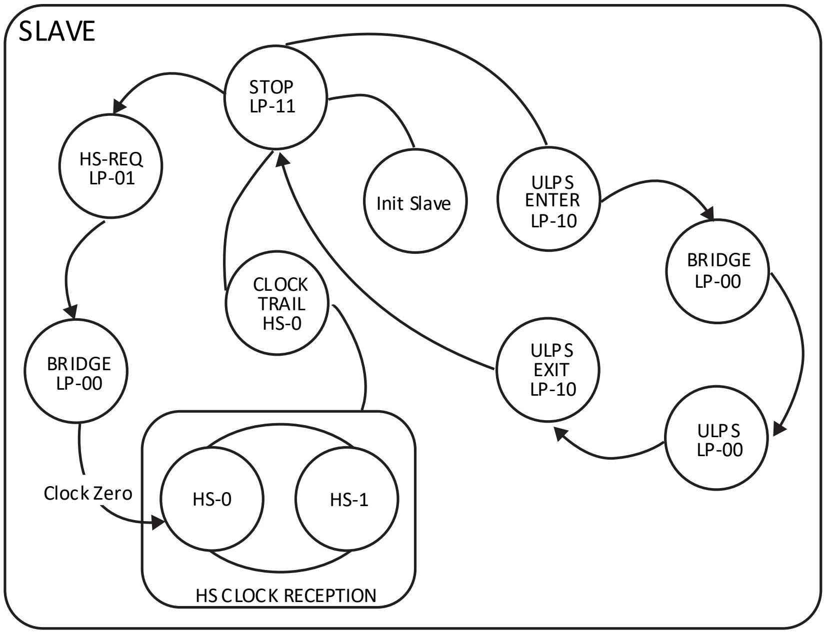


Figure 2.13. MIPI D-PHY Clock Lane Module State Diagram

Similarly, HS termination enable of data lanes becomes high after proper LP to HS transition is detected on data lane 0. A free-running byte clock is used for this function. Figure 2.14 shows the required LP to HS transition on data lanes as per MIPI D-PHY Specification version 1.1.

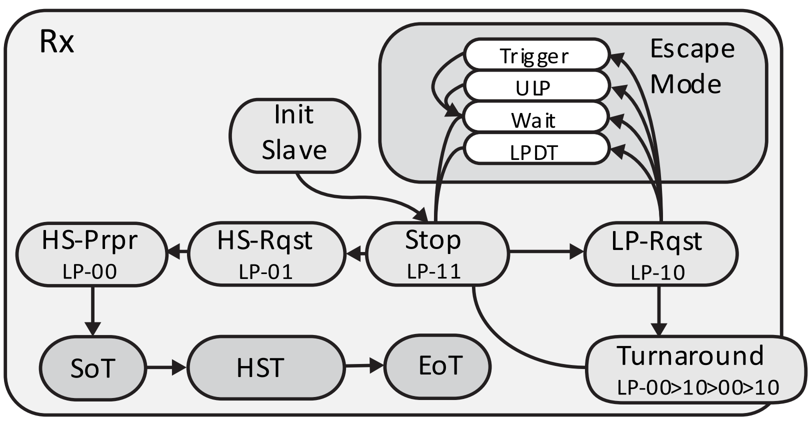


Figure 2.14. MIPI D-PHY Data Lane Module State Diagram

2.4.3. Capture Controller (Packet Parser)

This block parses the data bytes from D-PHY Common Interface Wrapper and detects short and long packets defined by MIPI DSI or MIPI CSI-2. This block extracts video data and other control parameters from the packets.

There are no signals from the external logic to this block to control the flow of output data. The interfacing logic must provide ample buffering to ensure the continuous flow of data from this submodule is transferred correctly.

2.5. Reset and Clocking

Asynchronous active low reset input (reset_n_i) is used as a system reset. To reset logic in continuous byte clock domain, system reset passes through synchronization registers to create asynchronous reset assertion and synchronous reset deassertion. The system reset input must be asserted for at least three times the byte clock cycle.

When MIPI D-PHY clock is continuous, it is expected to be in high-speed mode at power on of the device. The HS termination enable of clock lane is tied to VCC. Continuous byte clock is generated by the D-PHY IP. This output byte clock may be connected to the input freerunning clock clk_byte_fr_i.

An external clock input is required for detecting the LP to HS transition of clock (clk_lp_ctrl_i). This could be the same input free-running byte clock clk_byte_fr_i, as long as the clk_lp_ctrl_i clock period is less than the T_{LPX} (transmitted length of any low-power state period) with enough setup and hold margins.

Table 2.2 lists the frequency calculations. DCK refers to MIPI D-PHY clock frequency.

Table 2.2. Clock Frequency Calculations

Clock	Formula
D-PHY clock	DCK
Rx line rate	DCK × 2
Byte clocks (clk_byte_o and clk_byte_fr)	DCK / (Rx gear / 2)
LP Control (clk_lp_ctrl_i)	2 × (Rx D-PHY low-power mode frequency)

2.5.1. Clock Domains

Figure 2.15 shows the three clock domains within the D-PHY Rx Submodule.

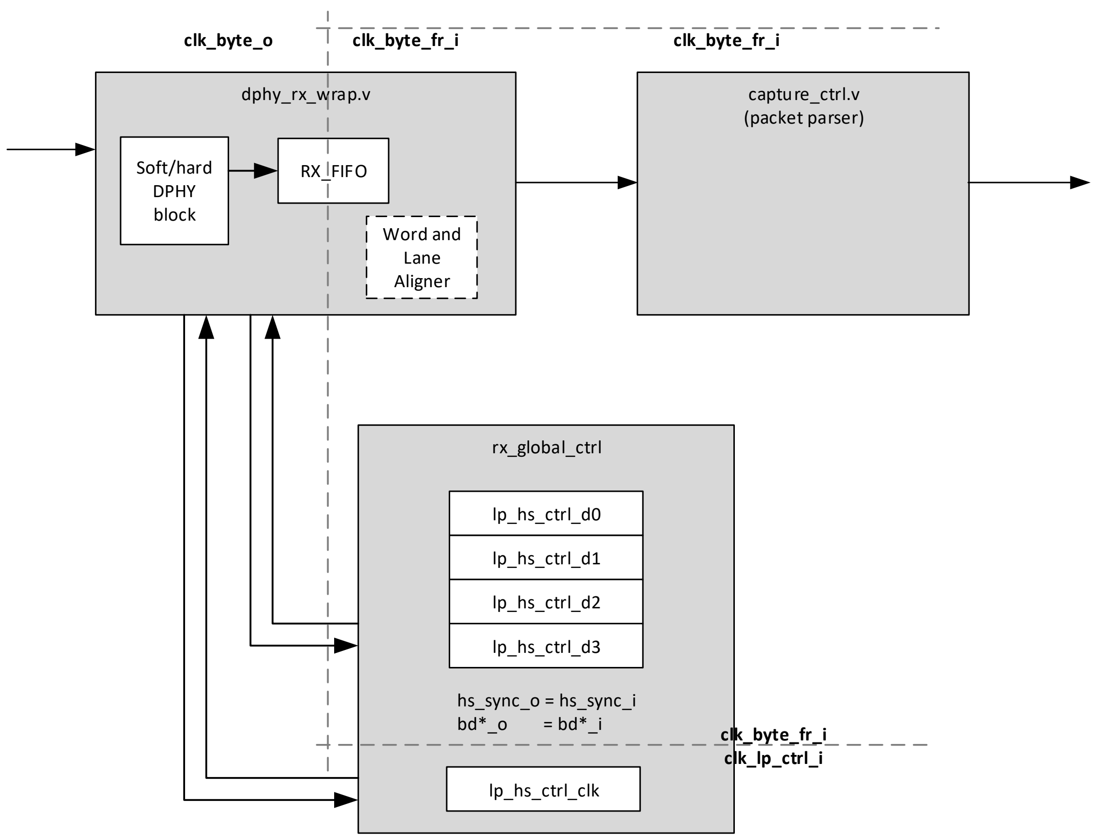


Figure 2.15. Clock Domain Diagram

The period of the clk_lp_ctrl_i should be smaller than the tLPX, with enough setup and hold time, to properly sample the Low Power state transitions. No need to drive this clock if the Rx clock mode is HS_ONLY.

The output signal clk_byte_o is non-continuous and is only active when the Rx D-PHY data lanes are in high-speed mode. This clock is used for latching the data bytes from the deserializer block.

3. Configuration Options

This section lists the settings used to configure the D-PHY Receiver Submodule.

Table 3.1. Configuration Options

User Interface Config Tab Sub-section	Parameter	Attribute	Options	Description
Receiver	Rx Interface	User-Input	DSI or CSI-2	Receive interface
	Number of Rx Lanes	User-Input	1, 2, 3, or 4	Number of MIPI D-PHY data lanes. 3-lane configuration is only valid when packet parser is disabled.
	Rx Gear	User-Input	8, 16	Rx gear is automatically computed based on selected configuration and D-PHY data rate.
	Rx D-PHY IP	User-Input	Hard D-PHY or Soft D-PHY	MIPI D-PHY Implementation. Soft D-PHY uses programmable MIPI I/O while Hard D-PHY implementation uses hardened logic blocks.
Clock	Rx Line Rate	User-Input	80 Mb/s – 1200 Mb/s (Hardened D-PHY and Gear 8) 80 Mb/s – 1200 Mb/s (Soft D-PHY and Gear 8) 160 Mb/s – 1500 Mb/s (Hardened D-PHY and Gear 16) 160 Mb/s – 1200 Mb/s (Soft D-PHY and Gear 16)	Bit rate per Rx data lane in high-speed mode. This is twice the D-PHY clock lane frequency.
	D-PHY Clock Frequency	Read-only	—	D-PHY clock frequency
	D-PHY Clock Mode	User-Input	Continuous or Non-continuous	Rx D-PHY is in Continuous clock mode when the D-PHY clock lanes are in high-speed mode. The termination is always enabled in this mode. The clock mode is non-continuous if the clock lane goes to low-power mode in between high-speed data transmission.
	Byte Clock Frequency	Read-only	—	Byte clock frequency
	Settle Period	User-Input	1–25	This is the number of byteclock period wherein the receiver ignores the data lane transitions starting from the detection of LP-00 state in the data lane0. This corresponds to the tHS-SETTLE in the MIPI Specification for D-PHY.
Output	Packet Parser	User-Input	Enabled or Disabled	Includes or excludes the packet parser in design.
IP Implementation	Soft Aligner Modules	User-Input	Enabled or Disabled	Includes or excludes the Word and Lane Aligner Module in the design.

User Interface Config Tab Sub-section	Parameter	Attribute	Options	Description
Settings	Lane Aligner FIFO Type	User-Input	EBR or LUT	Lane aligner FIFO implementation per lane. This has no change in functionality. Modify the implementation type depending on the resource requirement of the overall design.
Rx FIFO	Type	User-Input	EBR or LUT	Type of FIFO which is used to synchronize the parallel data from Byte Clock strobe.
	Depth	User-Input	16, 32, 64, 128, 256, 512, 1024	Depth of FIFO which is used to synchronize the parallel data from Byte Clock strobe.
	Mode	User-Input	“PINGPONG”, “SINGLE”, “QUEUE”	Implementation of FIFO which is used to synchronize the parallel data from Byte Clock strobe.
	Packet Delay	User-Input	0–1023	0 – Entire hs-transaction is buffered before FIFO is read. 1 – 1023 uses delay counter before FIFO is read.
	Counter Width	User-Input	1–12	Indicates maximum number of byte clock cycles in a high speed transaction.

4. IP Generation and Evaluation

This section provides information on how to generate Lattice CSI-2/DSI D-PHY Receiver Submodule IP using Lattice Diamond Clarity Designer and how to run simulation, synthesis, and hardware evaluation.

4.1. Licensing the IP

The CSI-2/DSI D-PHY Receiver Submodule IP license is available free of charge, but an IP-specific license is required to enable full, unrestricted use of the CSI-2/DSI D-PHY Receiver Submodule IP in a complete, top-level design.

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

You may download or generate the CSI-2/DSI D-PHY Receiver Submodule IP and fully evaluate it through functional simulation and implementation (synthesis, map, place, and route) without the IP license. The CSI-2/DSI D-PHY Receiver Submodule IP also supports Lattice IP hardware evaluation capability, see the Hardware Evaluation section for further details.

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

4.2. Getting Started

The CSI-2/DSI D-PHY Receiver Submodule IP is available for download from the Lattice IP Server using the Clarity Designer tool. The IP files are automatically installed using ispUPDATE technology in any customer-specified directory. After the IP has been installed, the IP is available in the Clarity Design user interface as shown in Figure 4.1.

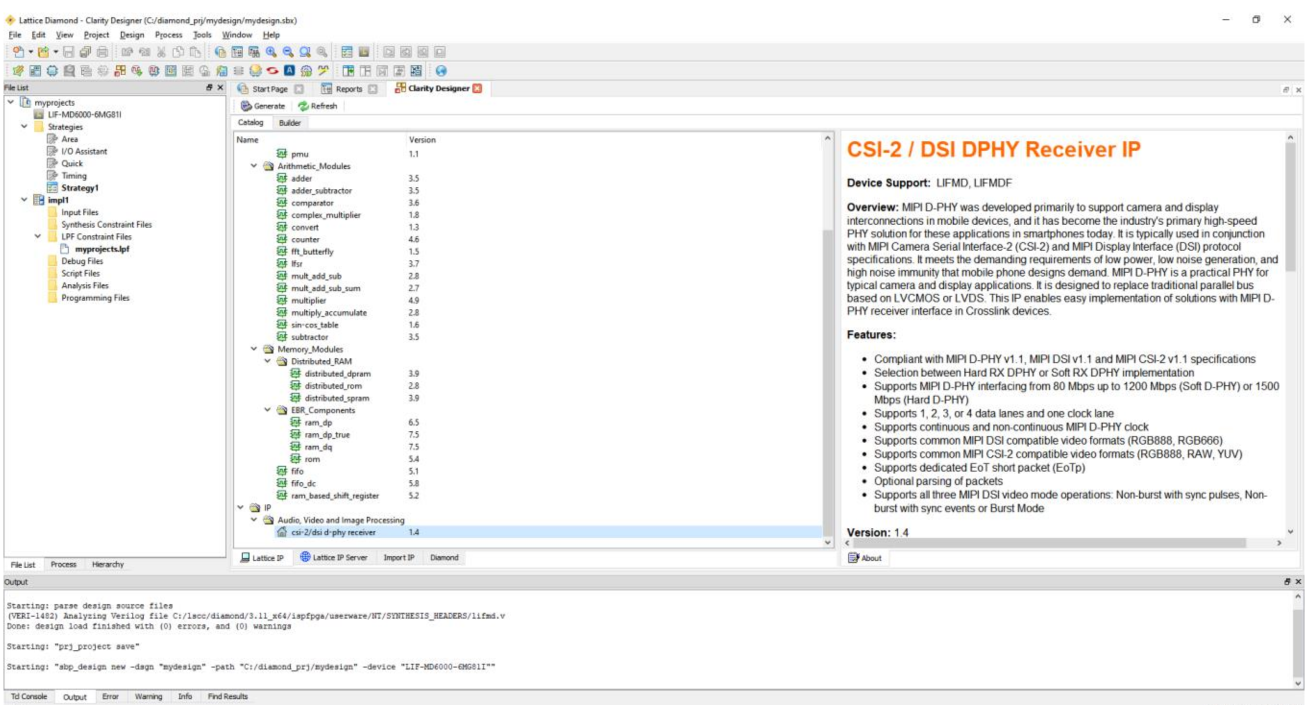


Figure 4.1. Clarity Designer Window

4.3. Generating IP in Clarity Designer

The Clarity Designer tool is used to customize modules and IPs and place them into the device architecture. The following describes the procedure for generating CSI-2/DSI D-PHY Receiver Submodule IP in Clarity Designer.

Clarity Designer is started from Lattice Diamond design environment.

To start Clarity Designer:

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

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

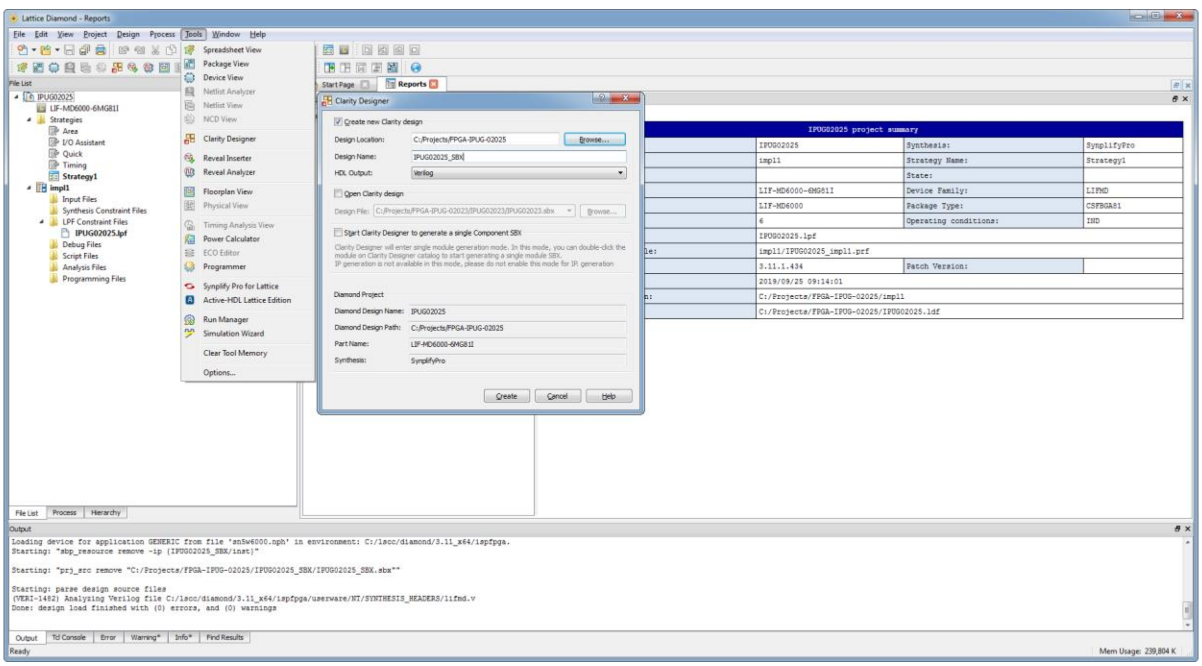


Figure 4.2. Starting Clarity Designer from Diamond Design Environment

To configure the CSI-2/DSI D-PHY Receiver Submodule IP in Clarity Designer:

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

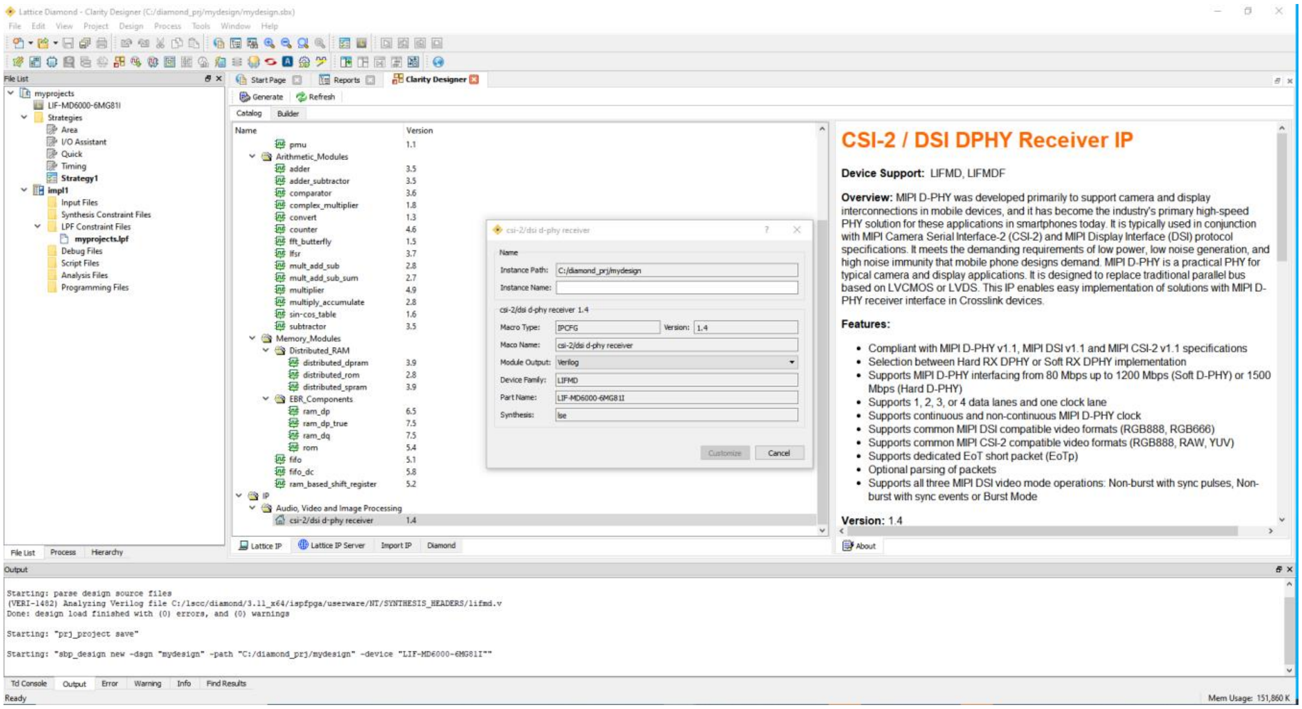


Figure 4.3. Configuring CSI-2/DSI D-PHY Receiver Submodule IP in Clarity Designer

2. Enter the **Instance Name**.
3. Click the **Customize** button. An IP configuration user interface is displayed as shown in Figure 4.4. From this dialog box, you can select the IP parameter options specific to your application.

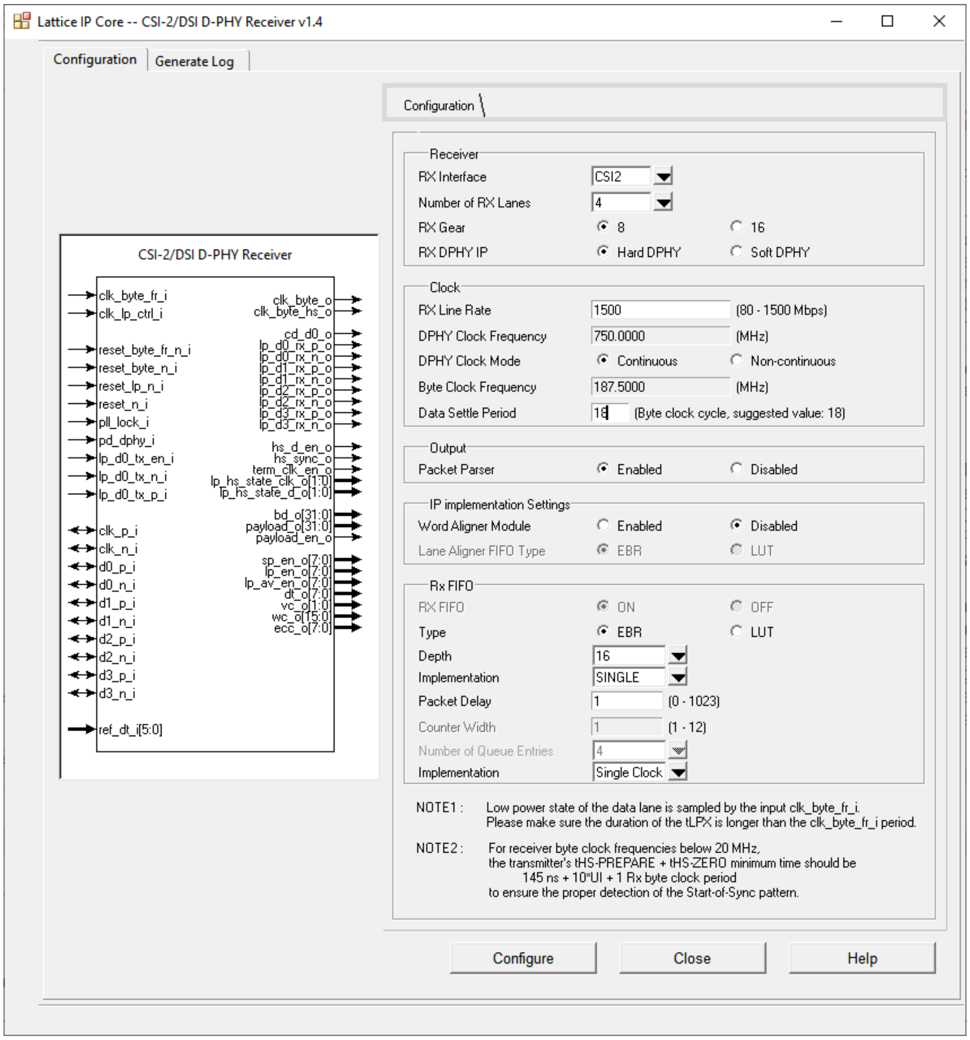
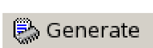


Figure 4.4. Configuration Tab in IP Interface

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

in the toolbox. Clarity Designer generates all the IPs and modules, and creates a top module to wrap them.

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

4.4. Generated IP Directory Structure and Files

Figure 4.5 shows the directory structure of the generated IP files.

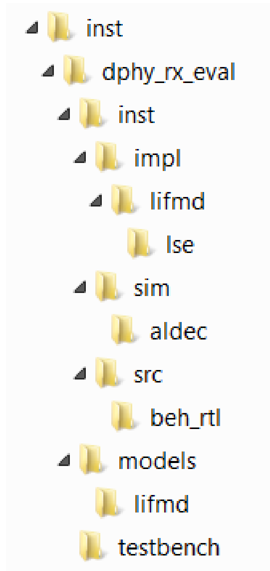


Figure 4.5. CSI-2/DSI D-PHY Receiver Submodule IP Directory Structure

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

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

Table 4.1. Files Generated by Clarity Designer

File	Description
<instance_name>.v	Verilog top-level module of CSI-2/DSI D-PHY Receiver Submodule IP used for both synthesis and simulation.
<instance_name>_*.v	Verilog submodules for simulation. Files that do not have equivalent black box modules are also used for synthesis.
<instance_name>_wrap.v	Add this in the input files and set as top level wrapper to evaluate the design up to Place and Route. Otherwise, the standalone generated sbx file fails mapping due to the number of ports exceeding the number of pins in the device.
<instance_name>_*_beh.v	Protected Verilog models for simulation.
<instance_name>_*_bb.v	Verilog black box modules for synthesis.
<instance_name>_*.ngo	User interface configured and synthesized modules for synthesis.
<instance_name>_params.v	Verilog parameters file that contains required compiler directives to successfully configure IP during synthesis and simulation.
<instance_name>.lpc	Lattice Parameters Configuration file. This file records all the IP configuration options set through Clarity Designer. It is used by the IP generation script to generate configuration-specific IP. It is also used to reload parameter settings in the IP user interface in Clarity Designer when it is being reconfigured.
<instance_name>_inst.v/vhd	Template for instantiating the generated soft IP top-level in another user-created top module.

All IP files are generated inside *<project_dir>* directory (test folder in Figure 4.5). The *<project_dir> is <design_location>**<design_name>**<instance_name>*, see the Generating IP in Clarity Designer section. A separate *<project_dir>* is created each time CSI-2/DSI D-PHY Receiver Submodule IP is created with a different IP instance name.

The *dphy_rx_eval* and subdirectories provide files supporting push-button IP evaluation through functional simulations, design implementation (synthesis, map) and hardware *evaluation*. Inside the *dphy_rx_eval* is the *<instance_name>* folder (inst folder in Figure 4.5) which contains protected behavioral files in *<instance_name>**src**beh_rtl* and a pre-built Diamond project in *<instance_name>**impl**<device_family>**<synthesis_tool>*\, where *<device_family>* can either be lifmd for CrossLink or lifmdf for CrossLinkPlus devices.

The *<instance_name>* is the IP instance name you specified in Clarity Designer. The simulation part of user evaluation provides testbench and test cases supporting RTL simulation for Active-HDL simulator under *testbench* folder. Separate directories located at *<project_dir>**dphy_rx_eval**<instance_name>**sim**aldec* are provided and contain specific pre-built simulation script files. See the Running Functional Simulation section below for details.

4.5. Running Functional Simulation

The generated IP package includes the behavioral models *(<instance_name>_*_beh.v)* provided in *<project_dir>**dphy_rx_eval**<instance_name>**src**beh_rtl* for functional simulation. The testbench files are provided in *<project_dir>**dphy_rx_eval**testbench*.

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

1. Create new project using Lattice Diamond for Windows.
2. Open **Active-HDL Lattice Edition** interface tool.
3. To customize the testbench parameters, edit the file **tb_setup_params.v** inside the *<project_dir>**<instance_name>**dphy_rx_eval**testbench* folder. See Table 4.2 for the list of valid testbench compiler directives.
4. Click **Tools**, then click **Execute Macro**.
5. Select the **<instance_name>_run.do** file inside the <project_dir>\<instance_name>\dphy_rx_eval\<instance_name>\sim\aldec folder.
6. Wait for the simulation to finish.
7. Input and output log files are saved in the **sim** directory.

Testbench parameters and directives can be modified by setting the define in the vlog command in the ***.do** file. Table 4.2 lists the testbench directives common for DSI and CSI-2 Rx type.

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

Directive	Description
NUM_FRAMES	Set the number of video frames.
NUM_LINES	Set the number of lines per frame.
VIRTUAL_CHANNEL	Set the virtual channel number.
DPHY_DEBUG_ON	Enable or disable debug messages. 0 – Debug messages disabled 1 – Debug messages enabled
DPHY_CLK	Set the D-PHY clock period (in ps).
REF_CLK	Override the default clk_byte_fr_i clock period for HS_LP clock mode.
FRAME_LPM_DELAY	Set the low-power mode delay between frames (in ps).
INIT_DELAY	Set the delay before data is transmitted to the design.

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

Table 4.3. Testbench Directives for D-PHY Timing Parameters

Directive	Description
DPHY_LPX	Set T-LPX (in ps).
DPHY_CLK_PREPARE	Set T-CLK-PREPARE (in ps).
DPHY_CLK_ZERO	Set T-CLK-ZERO (in ps).
DPHY_CLK_PRE	Set T-CLK-PRE (in ps).
DPHY_CLK_POST	Set T-CLK-POST (in ps).
DPHY_CLK_TRAIL	Set T-CLK-TRAIL (in ps).
DPHY_HS_PREPARE	Set T-HS-PREPARE (in ps).
DPHY_HS_ZERO	Set T-HS-ZERO (in ps).
DPHY_HS_TRAIL	Set T-HS-TRAIL (in ps).
DPHY_INIT	Set the T-INIT timing (in ps).

Table 4.4 is a list of testbench directives for DSI Rx type.

Table 4.4. Testbench Directives for DSI Rx Type

File	Description
LP_BLANKING	Drive low-power blanking. If this is not defined, the testbench drives HS data as blanking.
NON_BURST_SYNC_PULSE	DSI video modes
NON_BURST_SYNC_EVENTS	
BURST_MODE	
RGB888	DSI data types
RGB666	
RGB666_LOOSE	
DSI_VACT_PAYLOAD	Number of bytes of active pixels per line
DSI_HSA_PAYLOAD	Number of bytes of Horizontal Sync Active Payload (used for Non-burst sync pulse)
DSI_BLLP_PAYLOAD	Number of bytes of BLLP Payload (used for HS data blanking)
DSI_HBP_PAYLOAD	Number of bytes of Horizontal Back Porch Payload (used for HS data blanking, and in LP blanking for Non-burst sync pulse mode)
DSI_HFP_PAYLOAD	Number of bytes of Horizontal Front Porch Payload (used for HS data blanking, and in LP blanking for Non-burst sync pulse mode)
DSI_VSA_LINES	Number of Vertical Sync Active Lines
DSI_VBP_LINES	Number of Vertical Back Porch Lines
DSI_VFP_LINES	Number of Vertical Front Porch Lines
DSI_EOTP_ENABLE	Enable/disable transmission of EoTP packet. 0 – EoTP packet is disabled 1 – EoTP packet is enabled
DSI_LPS_BLLP_DURATION	Set the duration (in ps) for BLLP low-power state (used for LP blanking).
DSI_LPS_HBP_DURATION	Set the duration (in ps) for Horizontal Back Porch low-power state (used for LP blanking in Non-burst sync events and Burst mode).
DSI_LPS_HFP_DURATION	Set the duration (in ps) for Horizontal Front Porch low-power state (used for LP blanking in Non-burst sync events and Burst mode).

Table 4.5 is a list of testbench directives for CSI-2 Rx type.

Table 4.5. Testbench Directives for CSI-2 Rx Type

File	Description
CSI2_LPS_GAP	Set low-power state delay between HS transactions (in ps).
CSI2_NUM_PIXELS	Set the number of pixels per line.
CSI2_LS_LE_EN	Enable/disable D-PHY model transmission of line start and line end packets. 0 – No Line start and Line end packets 1 – Line start and Line end packets enable
RGB888 RAW8 RAW10 RAW12 YUV420_10 YUV420_10_CSPS YUV420_8 YUV420_8_CSPS LEGACY_YUV420_8 YUV422_10 YUV422_8	CSI-2 data types

4.6. Simulation Strategies

This section describes the simulation environment, which demonstrates basic MIPI D-PHY Rx functionality. Figure 4.6 shows the block diagram of simulation environment.

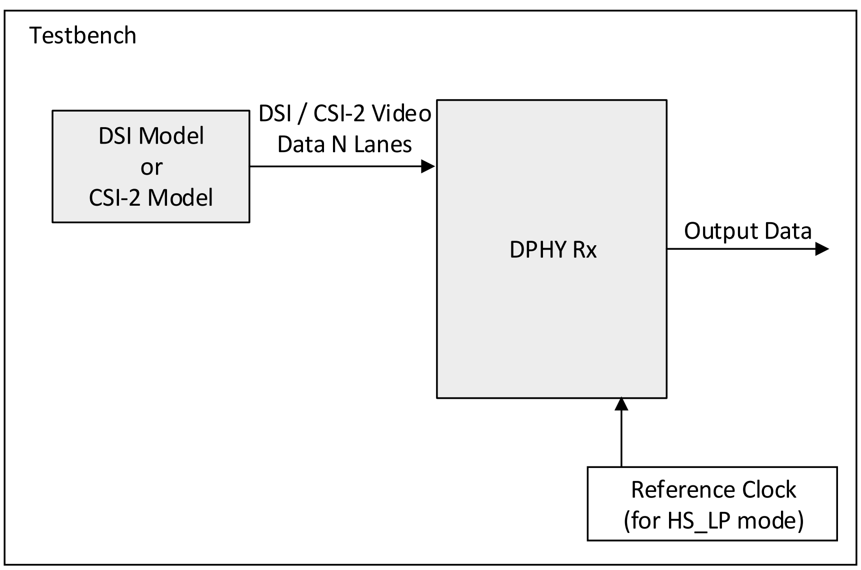


Figure 4.6. Simulation Environment Block Diagram

4.7. Simulation Environment

The simulation environment is made up of the DSI model instance if Rx type is DSI, or made up of CSI-2 model if Rx type is CSI-2. The instantiated model is connected to the D-PHY Rx IP core instance in the testbench. The DSI model or CSI-2 model is configured based from the D-PHY Rx IP core configurations and testbench configurations. Refer to testbench *tb_setup_params.v* file for details on how to set testbench parameters. The DSI or CSI-2 model transmits the video data to the CSI-2/DSI D-PHY Receiver Submodule IP core after reset. The testbench also transmits reference clock to the CSI-2/DSI D-PHY Receiver Submodule IP core, if clock mode is non-continuous (HS_LP). If clock mode is continuous (HS_ONLY), the clk_byte_hs_o is connected to clk_byte_fr_i.

The video data transmitted by the DSI model can be viewed in the waveform. See Figure 4.7.

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

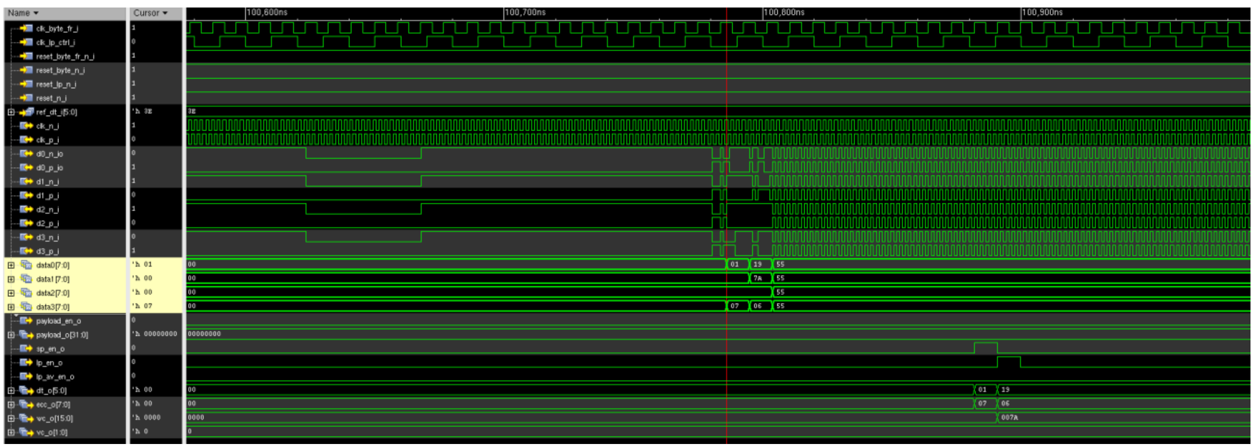


Figure 4.7. DSI Model Video Data

The video data transmitted by the CSI-2 model can be viewed in the waveform. See Figure 4.8.

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

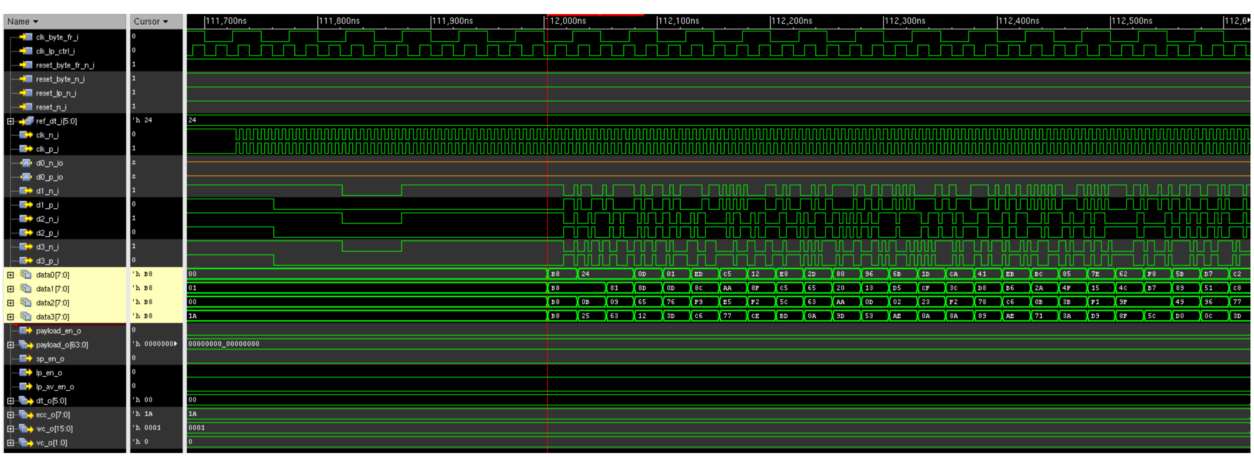


Figure 4.8. CSI-2 Model Video Data

4.8. Instantiating the IP

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

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

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

4.9. Synthesizing and Implementing the IP

In Clarity Designer, the Clarity Designer project file (.sbx) is added to Lattice Diamond as a source file after IP is generated. All required Verilog source files for implementation are invoked automatically. The IP can be directly synthesized, mapped and placed/routed in the Diamond design environment after the IP is generated. Note that default Diamond strategy (.sty) and default Diamond preference file (.lpf) are used. When using the .sbx approach, import the recommended strategy and preferences from
\<project_dir>\dphy_rx_eval\<instance_name>\impl\<device_family>\lse or
\<project_dir>\dphy_rx_eval\<instance_name>\impl\<device_family>\synplify directories and set them as active strategy and active preference file.

Push-button implementation of this IP with either Lattice Synthesis Engine (LSE) or Synopsys Synplify Pro RTL synthesis is supported via the pre-built Diamond project file *<instance_name>_top.ldf* located in
\<project_dir>\dphy_rx_eval\<instance_name>\impl\<device_family>\lse or
\<project_dir>\dphy_rx_eval\<instance_name>\impl\<device_family>\synplify directories.
To use the pre-built Diamond project file:

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

4.10. Hardware Evaluation

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

4.10.1. Enabling Hardware Evaluation in Diamond

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

4.11. Updating/Regenerating the IP

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

4.11.1. Regenerating an IP in Clarity Designer

To regenerate IP in Clarity Designer:

1. In the **Builder** tab, right-click the IP instance to be regenerated and select **Config** from the menu as shown in Figure 4.9.

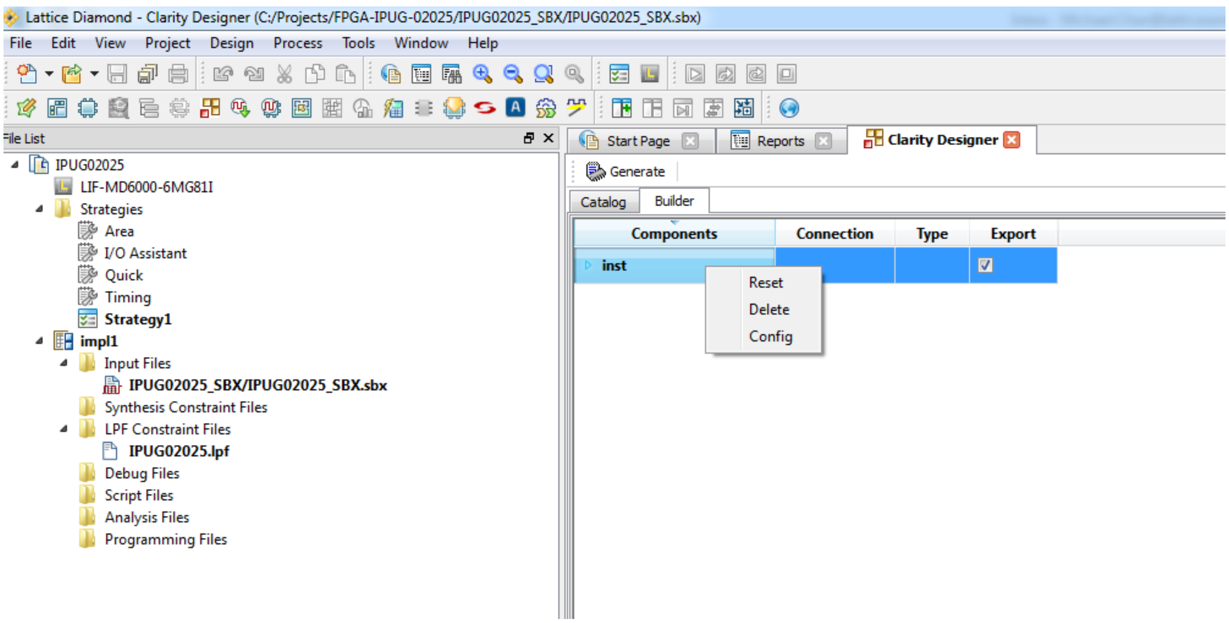
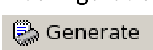


Figure 4.9. Regenerating IP in Clarity Designer

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

References

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

Software documentation:

- Clarity Designer User Manual
- Lattice Diamond User Guide

For further information on interface standards, refer to:

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

Technical Support Assistance

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

Appendix A. Resource Utilization

Table A.1 lists resource utilization for Lattice CrossLink FPGAs using the CSI-2/DSI D-PHY Receiver Submodule IP. The values shown below are based on map reports.

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

IP User-Configurable Parameters	Registers	Slices	LUTs	sysMEM EBRs	Programmable I/O	MIPI D-PHY	Target f_{MAX} (MHz)[2]	Actual f_{MAX} (MHz)[2]
CSI-2 Hard D-PHY Gear 16								
4-lane Non-continuous clock Packet parser disabled Word aligner disabled Lane aligner disabled	241	190	250	4	0	1	93.75	185.11
2-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner LUT FIFO	574	812	1183	2	0	1	93.75	230.52
1-lane Non-continuous clock Packet parser disabled Word aligner disabled Lane aligner disabled	145	140	184	1	0	1	93.75	179.86
4-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	708	1338	2023	8	0	1	93.75	184.98
2-lane Continuous clock Packet parser disabled Word aligner disabled Lane aligner disabled	168	148	191	2	0	1	93.75	207.9
1-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner disabled	312	408	594	1	0	1	93.75	193.94
CSI-2 Hard D-PHY Gear 8								
4-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	563	866	1256	6	0	1	150	207.64
2-lane Non-continuous clock Packet parser enabled Word aligner disabled Lane aligner disabled	300	286	403	1	0	1	150	225.53
1-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner disabled	281	331	461	1	0	1	150	193.01

IP User-Configurable Parameters	Registers	Slices	LUTs	sysMEM EBRs	Programmable I/O	MIPI D-PHY	Target f_{MAX} (MHz)[2]	Actual f_{MAX} (MHz) [2]
4-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	554	857	1245	6	0	1	150	—
2-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	354	489	708	3	0	1	150	200.92
1-lane Continuous clock Packet parser disabled Word aligner disabled Lane aligner disabled	120	118	164	1	0	1	150	208.64
CSI-2 Soft D-PHY Gear 16								
4-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner LUT FIFO	511	1079	1573	8	10	0	75	164.17
2-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	470	668	1043	2	6	0	75	200.8
1-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner disabled	262	361	547	1	4	0	75	211
4-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	503	1073	1566	8	10	0	75	175.87
2-lane Continuous clock Packet parser disabled Word aligner enabled Lane aligner EBR FIFO	225	450	671	4	6	0	75	218.82
1-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner disabled	105	123	175	1	4	0	75	245.28
CSI-2 Soft D-PHY Gear 8								
4-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	523	783	1166	2	10	0	150	218.2

IP User-Configurable Parameters	Registers	Slices	LUTs	sysMEM EBRs	Programmable I/O	MIPI D-PHY	Target f_{MAX} (MHz)[2]	Actual f_{MAX} (MHz)[2]
2-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner LUT FIFO	304	442	661	3	6	0	150	232.07
1-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner disabled	247	297	429	1	4	0	150	216.66
4-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	450	736	1109	6	10	0	150	202.72
2-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	223	346	496	2	6	0	150	258.6
1-lane Continuous clock Packet parser disabled Word aligner enabled Lane aligner disabled	238	290	420	1	4	0	150	235.18
DSI Hard D-PHY Gear 16								
4-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	769	1505	2265	8	0	1	93.75	197.71
2-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	517	779	1166	4	0	1	93.75	169.32
1-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner disabled	322	426	622	1	0	1	93.75	200.6
4-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	760	1496	2254	8	0	1	93.75	187.62
2-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	508	770	1155	4	0	1	93.75	200.2

IP User-Configurable Parameters	Registers	Slices	LUTs	sysMEM EBRs	Programmable I/O	MIPI D-PHY	Target f_{MAX} (MHz)[2]	Actual f_{MAX} (MHz)[2]
1-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner disabled	313	417	611	1	0	1	93.75	174.87
DSI Hard D-PHY Gear 8								
4-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	571	874	1286	6	0	1	150	231.91
2-lane Non-continuous clock Packet parser enabled Word aligner disabled Lane aligner disabled	301	290	414	1	0	1	150	225.38
1-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner disabled	279	326	444	1	0	1	150	249.63
4-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	526	865	1275	6	0	1	150	245.22
2-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	355	493	719	3	0	1	150	224.72
1-lane Continuous clock Packet parser disabled Word aligner disabled Lane aligner disabled	120	118	164	1	0	1	150	208.64
DSI Soft D-PHY Gear 16								
4-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner LUT FIFO	687	1245	1877	4	10	0	75	211.33
2-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	411	654	1025	4	6	0	75	243.6
1-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner disabled	261	370	564	1	4	0	75	205.8

IP User-Configurable Parameters	Registers	Slices	LUTs	sysMEM EBRs	Programmable I/O	MIPI D-PHY	Target f_{MAX} (MHz)[2]	Actual f_{MAX} (MHz)[2]
4-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	550	158	1772	8	10	0	75	182.55
2-lane Continuous clock Packet parser disabled Word aligner enabled Lane aligner EBR FIFO	255	450	671	4	6	0	75	208.68
1-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner disabled	252	363	555	1	4	0	75	231.8
DSI Soft D-PHY Gear 8								
4-lane Non-continuous clock Packet parser disabled Word aligner enabled Lane aligner EBR FIFO	288	550	800	6	10	0	150	234.03
2-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner LUT FIFO	335	466	696	1	6	0	150	257.73
1-lane Non-continuous clock Packet parser enabled Word aligner enabled Lane aligner disabled	254	293	416	1	4	0	150	246.73
4-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner EBR FIFO	456	741	1136	6	10	0	150	235.57
2-lane Continuous clock Packet parser enabled Word aligner enabled Lane aligner LUT FIFO	326	459	687	1	6	0	150	207.34
1-lane Continuous clock Packet parser disabled Word aligner enabled Lane aligner disabled	116	152	214	1	4	0	150	243.49

Notes:

1. The performance and utilization data target an LIF-MD6000-6MG81I device with -6 speed grade using Lattice Diamond 3.9 and Lattice Synthesis Engine. Performance may vary when using a different software version or targeting a different device density or speed grade within the CrossLink family.
2. The target f_{MAX} is the target byte clock frequency of each configuration. Actual f_{MAX} may vary depending on the complete top-level design.

3. The distributed RAM utilization is accounted for in the total LUT4 utilization. The actual LUT4 utilization is distribution among logic, distributed RAM, and ripple logic.

Appendix B. What is Not Supported

The MIPI D-PHY Rx submodule IP does not support the following features:

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

The CSI-2/DSI D-PHY Receiver Submodule IP has the following design limitations:

- Maximum value of word count in a long packet is 16'hFFF5.
- Maximum fabric speed is 150 MHz. Configurations that result in byte clock frequencies greater than 150 MHz might encounter timing violations.
- Ports for Low-Power Data Transmissions (LPDT) communications are provided. However, these are not fully tested in simulations.
- The design supports all DSI and CSI-2 compatible formats. However, only common video DSI and CSI-2 formats are used to simulate the design:
  - DSI – RGB888, RGB666
  - CSI-2 – RGB888, RAW, YUV

See Table 4.4 and Table 4.5 for the list of data types used in simulation.

- The RX_FIFO QUEUE type has been tested in limited timing configurations. You are advised to adjust the queue settings or explore the other types of RX FIFO in case of errors.
- The Settle Period parameter is in unit of clk_byte_fr_i cycles. This makes the granularity coarser in lower clock frequencies. It is advised to check the tHS-PREPARE and the tHS-ZERO timing of the D-PHY source.

Revision History

Revision 1.5, IP Version 1.5, November 2021

Section	Change Summary
Configuration Options	Updated the line rate range from “160 Mb/s – 1500 Mb/s” to “80 Mb/s – 1200 Mb/s” for when Gear 8 is selected in the “Options” for “Rx Line Rate” in Table 3.1. Configuration Options.
Introduction	Changed 'Supports MIPI D-PHY interfacing from 160 Mb/s up to 1.5 Gb/s (Hard D-PHY)' to 'Supports MIPI D-PHY interface from 80 Mb/s up to 1.5 Gb/s (Hardened D-PHY / Gear 16)' in Section 1.2. Features.

Revision 1.4, IP Version 1.4, July 2020

Section	Change Summary
Introduction	Updated Table 1.1.
Appendix A. Resource Utilization	Updated Table A.1.

Revision 1.3, IP Version 1.4, March 2020

Section	Change Summary
Introduction	Updated Table 1.1.
Functional Description	• Updated Table 2.1. • Updated Figure 2.3 in Submodule without Packet Parser section. • Updated Figure 2.5 in Submodule with Packet Parser section. • Updated content of D-PHY Common Interface Wrapper section. • Updated Figure 2.12 in Clock Domains section.
IP Generation and Evaluation	• Updated Figure 4.1, Figure 4.3, and Figure 4.4. • Updated Table 4.1 in Generated IP Directory Structure and Files section.
Appendix B. What is Not Supported	Updated section.

Revision 1.2, IP Version 1.3, October 2019

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

Revision 1.1, IP Version 1.2, March 2019

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

Revision 1.0, IP Version 1.0, July 2017

Section	Change Summary
All	Initial release.

www.latticesemi.com