MIPI CSI/DSI IP

IP Version: v4.0.0

User Guide

FPGA-IPUG-02321-1.0

April 2026

Disclaimers

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

Inclusive Language

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

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
ASCII	American Standard Code for Information Interchange
AXI4-Stream	Advanced eXtensible Interface 4-Stream
B2P	Byte-to-Pixel
BPC	Bits per Color
BPP	Bits per Pixel
CPP	Color per Pixel
CRC	Cyclic Redundancy Check
CSI-2	Camera Serial Interface-2
D-PHY	Digital PHY (Physical Layer)
DDR	Double Data Rate (Serializer/Deserializer)
DSI-2	Display Serial Interface-2
EBR	Embedded Block Random Access Memory (RAM)
ECC	Error Correction Code
EoT	End of Transmission
FAQ	Frequently Asked Questions
FIFO	First In First Out
FPGA	Field-Programmable Gate Array
FSM	Finite State Machine
GUI	Graphical User Interface
HDL	Hardware Description Language
HS	High-Speed
IP	Intellectual Property
JTAG	Joint Test Action Group (debug interface)
LP	Low Power
LSE	Lattice Synthesis Engine
LUT	Look-up Table
MAP	Mapping Stage During FPGA Implementation
MIPI	Mobile Industry Processor Interface
P2B	Pixel-to-Byte
PDC	Physical Design Constraint
PHY	Physical Layer
PLL	Phase-Locked Loop
PPC	Pixel per Clock
RTL	Register Transfer Level
RX	Receiver
SDC	Synopsys Design Constraints
SOF	Start of Frame
SoT	Start of Transmission
TX	Transmitter
ULPS	Ultra Low Power State
UVSI	Unified Video Streaming Interface
VC	Virtual Channel
VCX	Virtual Channel Extension

1. Introduction

The Lattice™ MIPI® CSI/DSI IP provides a flexible solution for transmission and reception of high-speed video data over a Mobile Industry Processor Interface (MIPI)-compatible D-PHY interface in FPGA designs. MIPI enables high-speed, low-power video transmission between components like cameras and displays, and is widely used in smartphones, automotive systems, industrial equipment, and IoT devices.

1.1. Overview of the IP

The Lattice MIPI CSI/DSI IP transmitter supports Camera Serial Interface-2 (CSI-2) protocol while the receiver supports CSI-2 and Display Serial Interface-2 (DSI-2) protocols. The IP operates in both high-speed and low power modes and supports up to 1,500 Mbps per lane with Soft D-PHY implementation.

In transmitter mode, video data is received from the upstream pipeline at Unified Video Streaming Interface (UVSI), converted through the Pixel-to-Byte (P2B) block into byte-aligned data, and processed by the controller. This data is then serialized and transmitted through the PHY for seamless delivery to the connected receiver.

In receiver mode, incoming video data is received through the PHY, processed by the controller, and converted through the Byte-to-Pixel (B2P) block into UVSI packets for seamless integration with downstream video pipelines.

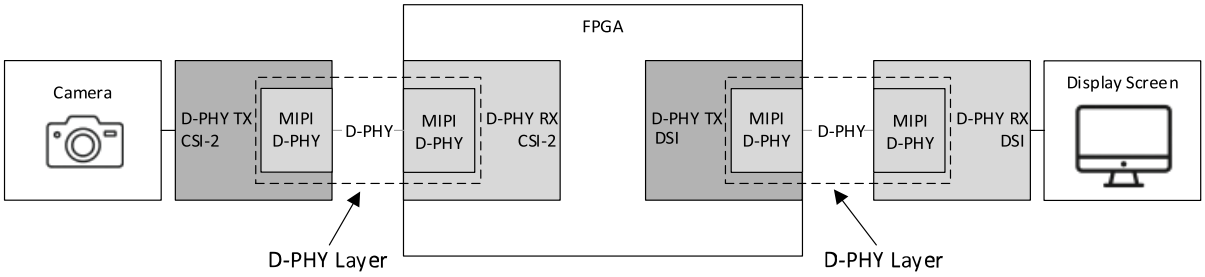

Figure 1.1. MIPI CSI/DSI IP Overview

1.2. Quick Facts

Table 1.1. Summary of the MIPI CSI/DSI IP

IP Requirements	Supported Devices	CrossLink™-NX, Certus™-NX, CertusPro™-NX, MachXO5™-NX
	IP Changes[1]	Refer to the MIPI CSI/DSI IP Release Notes (FPGA-RN-02112).
Resource Utilization	Supported User Interface	UVSI AXI4-Stream
	Resources	Refer to the Resource Utilization section.
Design Tool Support	Lattice Implementation	IP Core v4.0.0 – Lattice Radiant™ Software 2025.2 SP1
	Synthesis	Synopsys® Synplify Pro for Lattice
	Simulation	Refer to the Lattice Radiant software user guide for the list of supported simulators.

Note:

1. In some instances, the IP may be updated without changes to the user guide. This user guide may reflect an earlier IP version but remains fully compatible with the later IP version. Refer to the IP Release Notes for the latest updates.

1.3. IP Support Summary

Table 1.2. MIPI CSI/DSI IP Support Readiness

Device Family	PHY Direction	Protocol Mode	Line Rate Per Lane (Mbps)	Radiant Timing Model
CrossLink-NX, Certus-NX, CertusPro-NX, MachXO5-NX	TX	CSI-2	160–1500	Final
	RX	CSI-2 and DSI-2	80–1500	Final

1.4. Features

Key features of the MIPI CSI/DSI IP include:

- Supports MIPI D-PHY v2.1 with 1, 2, or 4 data lanes and 1 clock lane.
- Supports Soft D-PHY bandwidth up to 1.5 Gbps per lane, for a total bandwidth of 6 Gbps. See Table 1.2.
- Supports continuous and non-continuous MIPI D-PHY clock operation.
- Supports multiple color formats:
  - RGB888
  - RGB565
  - RAW10
  - RAW12
- Supports UVSI for pixel-domain transmission.

1.4.1. MIPI CSI/DSI IP on Transmitter Mode

- Supports MIPI CSI-2 v1.2.
  - Supports generation of Frame Start and Frame End with frame count.
  - Optional support for the generation of Line Start and Line End with optional Line Count.
- Supports 1, 2, and 4 pixels per clock (PPC).

1.4.2. MIPI CSI/DSI IP on Receiver Mode

- Supports MIPI CSI-2 v1.2 and DSI-2 v1.3.
- Supports all MIPI DSI video modes:
  - Non-burst mode with sync pulses.
  - Non-burst mode with sync events.
  - Burst mode.
- Supports 1, 2, and 4 pixels per clock.

1.5. Licensing and Ordering Information

The MIPI CSI/DSI IP is provided at no additional cost with the Lattice Radiant software.

1.6. Minimum Device Requirements

Refer to the Resource Utilization section for the minimum required resource to instantiate this IP.

1.7. Naming Conventions

1.7.1. Nomenclature

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

1.7.2. Signal Names

Signal names that end with:

- *_n* are active low signals (asserted when value is logic 0)
- *_i* are input signals
- *_o* are output signals
- *_io* are bidirectional signals

1.7.3. Attribute Names

Attribute names in this document are formatted in title case and italicized (*Attribute Name*).

2. Functional Description

2.1. IP Architecture Overview

The following figures show the architecture blocks and data flow in the MIPI CSI/DSI IP.

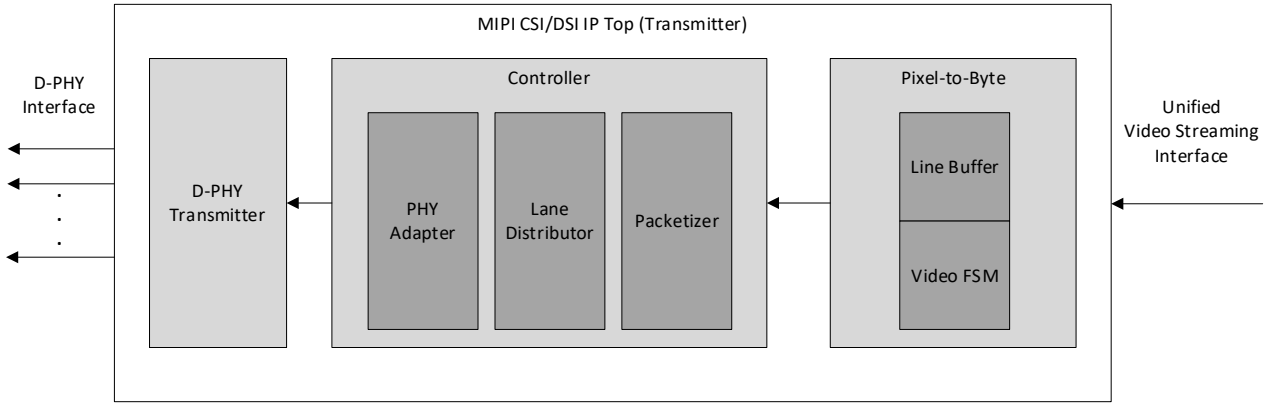

Figure 2.1. MIPI CSI/DSI IP Transmitter Core Block Diagram

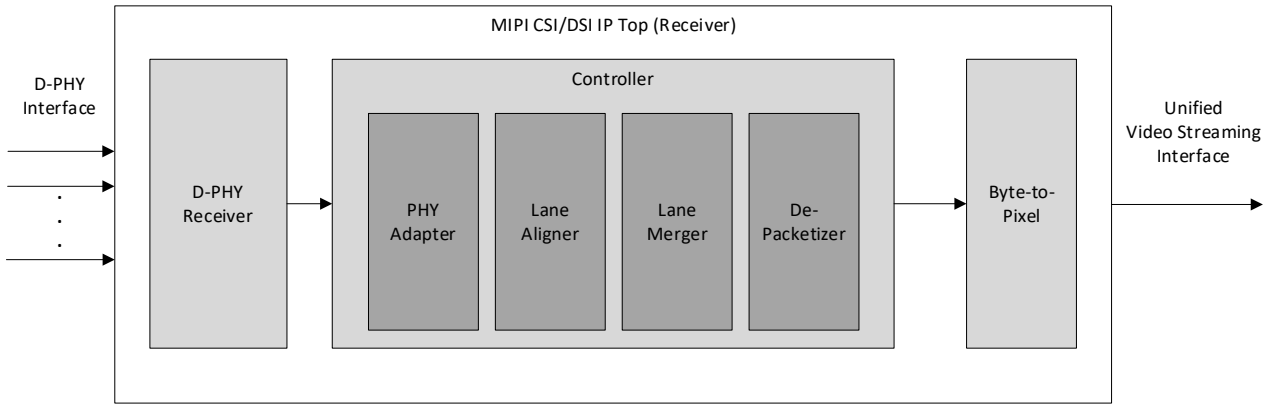

Figure 2.2. MIPI CSI/DSI IP Receiver Core Block Diagram

The MIPI CSI/DSI IP includes the following layers:

- Transmitter
  - PHY block
  - Controller block
  - Pixel-to-Byte block
    - Video FSM
- Receiver
  - PHY block
  - Controller block
  - Byte-to-Pixel block

2.2. Clocking

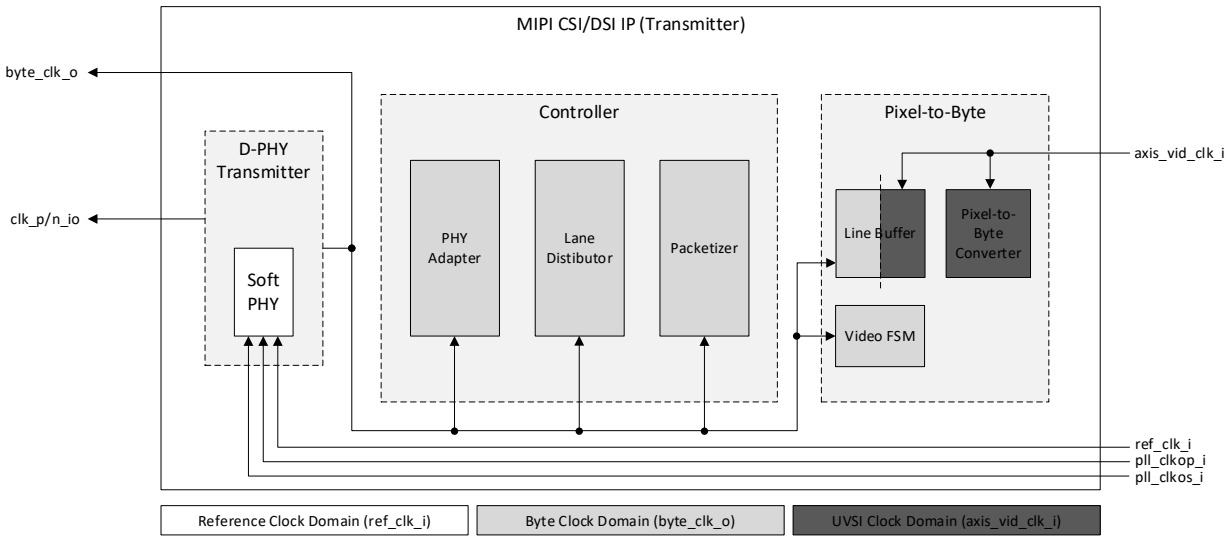

Figure 2.3. MIPI CSI/DSI IP Transmitter Clock Domain Block Diagram

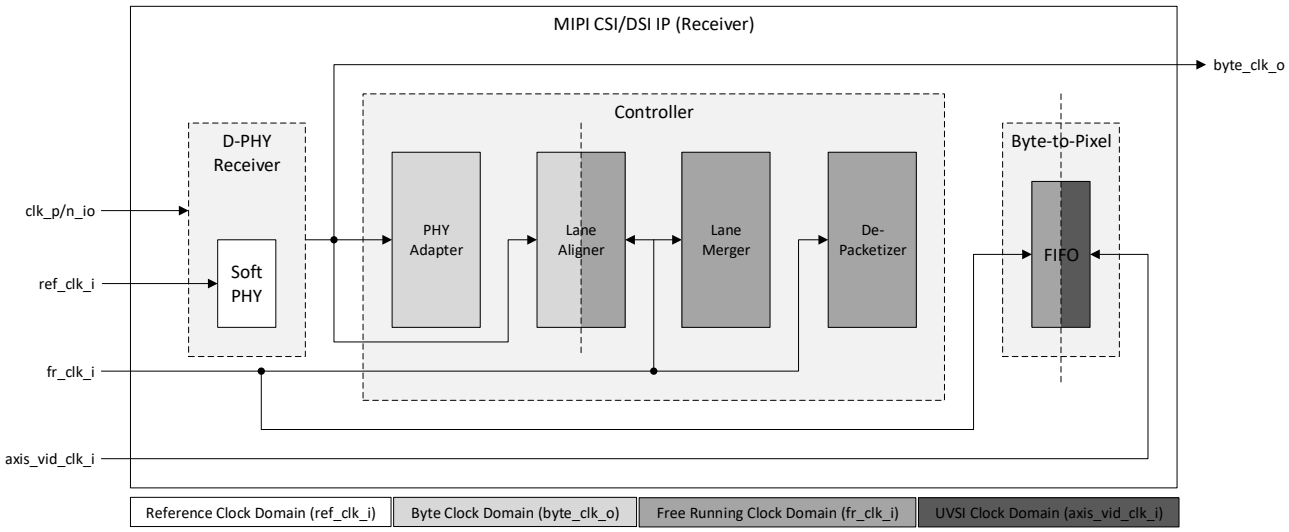

Figure 2.4. MIPI CSI/DSI IP Receiver Clock Domain Block Diagram

2.2.1. MIPI CSI/DSI Transmitter Clocking Overview

Clocks shown in Figure 2.3 are typically generated using an external PLL. For more details of each clock, refer to the Signal Description section.

Table 2.1. MIPI CSI/DSI Transmitter Operating Frequency Details and Requirements

Clock	Operating Frequency Details and Requirements
ref_clk_i	The operating frequency of this clock can be set on the *Reference Clock Frequency (MHz)* attribute. This is a low speed, continuously running input clock.
pll_clkop_i	This is the 0-degree phase output of an external PLL. This clock depends on the *Line Rate Per*

Clock	Operating Frequency Details and Requirements
	Lane (Mbps). Example: pll_clkop_i = (*Line Rate Per Lane (Mbps)*/2) For a configured line rate of 1000 Mbps per lane pll_clkop_i = (1000/2) = 500 MHz
pll_clkos_i	This clock operates at the same frequency as pll_clkop_i, but 90-degree phase shifted.
byte_clk_o[1,2]	The frequency of this clock can be calculated as: *Actual Byte Clock Frequency (MHz)* = *Line Rate Per Lane (Mbps)/PHY Data Width.* Byte clock is generated by gearing down pll_clkop_i and operates as a divided version of the PLL output clock.
axis_vid_clk_i	The video clock frequency must be selected such that the average input pixel rate does not exceed the effective throughput of the MIPI transmitted over time. When the video clock is slower than the instantaneous line data rate, sufficient horizontal blanking and/or line buffer depth must be provided to prevent buffer overflow. If these conditions are not met, the IP asserts backpressure on UVSI.

Notes:

1. The maximum data rate depends on the family, package, and performance grade of the device. Check the device data sheet for more information.
2. For edge clock sharing ports, refer to the MIPI CSI/DSI Transmitter Edge Clock Sharing section.

2.2.2. MIPI CSI/DSI Receiver Clocking Overview

Input clocks shown in Figure 2.4 are typically generated using an external PLL, while the output clock, byte_clk_o, is generated by an FPGA clock divider element. For more details of each clock, refer to the Signal Description section.

Table 2.2. MIPI CSI/DSI Receiver Operating Frequency Details and Requirements

Clock	Operating Frequency Details and Requirements
ref_clk_i	The operating frequency of this clock can be set on the *Reference Clock Frequency (MHz)* attribute with a range of 60 to 200 MHz.
byte_clk_o	The frequency of this clock can be calculated as: *Actual Byte Clock Frequency (MHz)* = *Line Rate Per Lane (Mbps)/PHY Data Width.*
fr_clk_i	The operating frequency of this clock can be set to greater than or equal to *Actual Byte Clock Frequency (MHz),* with a minimum value of 60 MHz.
axis_vid_clk_i	The operating frequency must be computed as follows, with minimum of *Actual Byte Clock Frequency (MHz):* axis_vid_clk_i_{MHz_TEMP} = ((*Number of D-PHY Lanes* × *PHY Data Width*) / (*Number of Pixel per Clock (PPC)* × BPP[1])) × *Actual Byte Clock Frequency (MHz)* If axis_vid_clk_i_{MHz_TEMP} < *Actual Byte Clock Frequency (MHz)*: • axis_vid_clk_i_{MHz} ≥ *Actual Byte Clock Frequency (MHz)* Else: • axis_vid_clk_i_{MHz} ≥ axis_vid_clk_i_{MHz_TEMP} Example 1: *Line Rate Per Lane (Mbps)* = 1,500 *PHY Data Width* = 8 *Number of D-PHY Lanes* = 4 *Number of Pixel per Clock (PPC)* = 4 *Enable RGB888 Color Format* = checked (BPP = 24) *Actual Byte Clock Frequency (MHz)* = 187.5 MHz axis_vid_clk_i_{MHz_TEMP} = ((4 × 8) / (4 × 24)) × 187.5 = 62.5 MHz As 62.5 MHz < 187.5 MHz, axis_vid_clk_i_{MHz} ≥ 187.5 MHz

Clock	Operating Frequency Details and Requirements
	Example 2: *Line Rate Per Lane (Mbps)* = 500 *PHY Data Width* = 8 *Number of D-PHY Lanes* = 4 *Number of Pixel per Clock (PPC)* = 1 *Enable RAW10 Color Format* = checked (BPP = 10) *Actual Byte Clock Frequency (MHz)* = 62.5 MHz axis_vid_clk_i_{MHz_TEMP} = ((4 × 8) / (1 × 10)) × 62.5 = 200 MHz As 200 MHz > 62.5 MHz, axis_vid_clk_iMHz ≥ 200 MHz

2.3. Reset

Resets of the IP are defined in the Signal Description section.

2.3.1. Reset Overview

When the clocks are generated by an external PLL, it is recommended to keep all associated resets active until the PLL achieves lock and the input clocks are stable.

There is no required reset sequence. However, all resets must be asserted together if a reset occurs during an active transaction. This ensures that any residual data inside internal FIFOs or pipelines is completely flushed. Failure to reset all relevant domains simultaneously during an in-transaction reset may leave stale data in the data path, which can lead to an IP hang or unexpected behavior.

2.4. User Interfaces

The table below lists the available user interface and protocols used on the MIPI CSI/DSI IP.

Table 2.3. User Interfaces and Supported Protocols

User Interface	Supported Protocols	Description
Device receiver	Unified Video Streaming Interface	This MIPI CSI/DSI transmitter interface accepts video data in the pixel domain and follows the standard AXI4-Stream protocol for transferring pixel information into the transmitter pipeline.
Device transmitter	Unified Video Streaming Interface	This MIPI CSI/DSI receiver interface transmits data in pixel domain and follows the standard AXI4-Stream protocol.

2.4.1. Unified Video Streaming Interface (UVSI)

The UVSI transfers video data in the pixel domain and complies with the AXI4-Stream protocol. For the complete list of interface signals, refer to the Signal Description section.

In UVSI, the first transmission of every frame is always the frame boundary, indicated by the assertion of the Start of Frame (SOF) signal. This transition marks the beginning of the valid video-frame data. The data flow strictly follows the AXI4-Stream handshaking requirements, where the source must assert TVALID only when valid data is available, and the sink must assert TREADY only when the sink is ready to accept data. A data beat is transferred only when both TVALID and TREADY are high, ensuring proper flow control and preventing data loss or corruption.

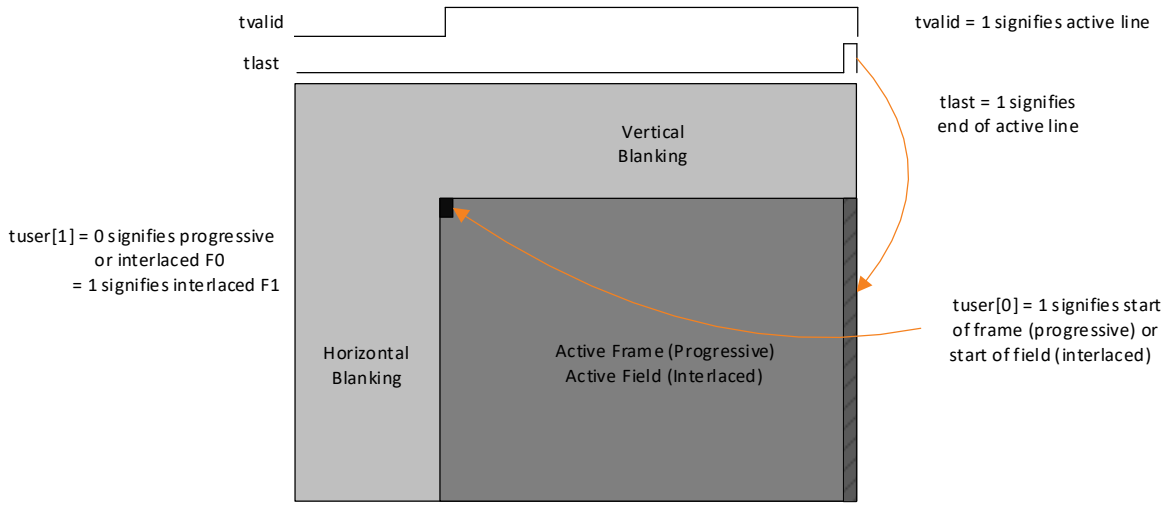

Figure 2.5. UVSI to Video Frame Timing Mapping

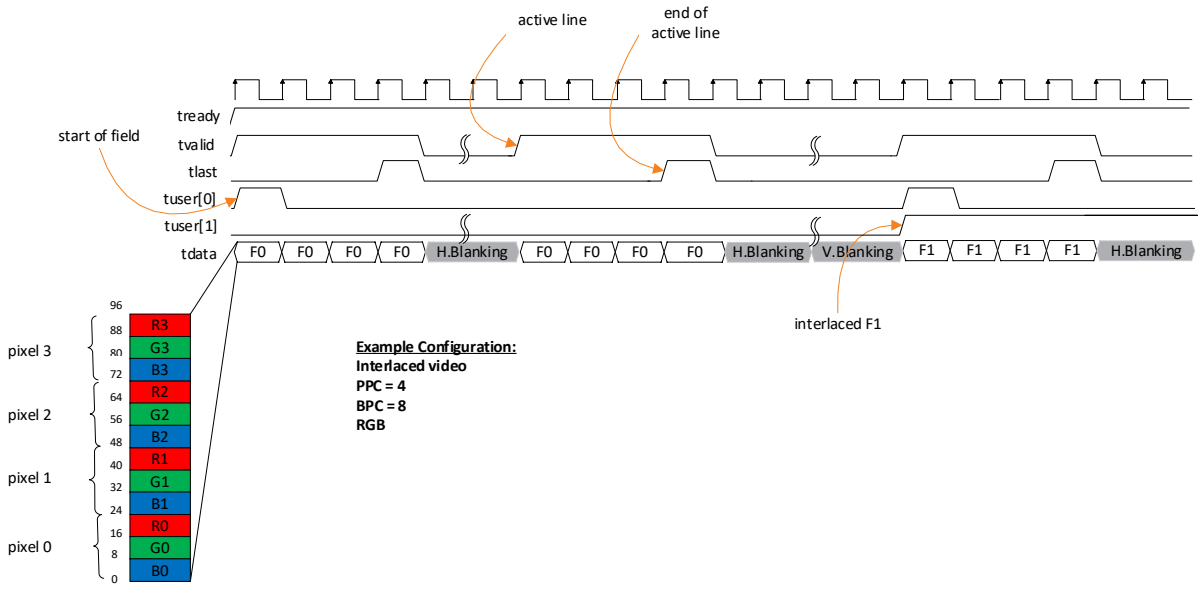

Figure 2.6. Unified Video Streaming Timing Diagram

The fundamental signaling rules of UVSI are consistent for both the TX and RX paths. Data movement is governed by the same AXI4-Stream-based handshake, ensuring reliable and flow-controlled transfer between the source and the sink. These common rules ensure predictable behavior whether the IP is transmitting or receiving video frames.

2.4.1.1. TDATA Mapping

Key definitions in TDATA mapping are as follows:

- Colors per Pixel (CPP)—Number of color components per pixel. For example, 1 for RAW, 3 for RGB.
- Bits per Color (BPC)—Configured bit width for each color component.
- Bits per Pixel (BPP)—Total bits per pixel = CPP × BPC.
- TDATA width per pixel—Width of UVSI TDATA allocated for one pixel, rounded up to the nearest byte boundary = ceil(BPP/8) × 8.
- TDATA_WIDTH—Total width of the AXI4-Stream TDATA bus for all pixels in a clock cycle = TDATA width per pixel × PPC.

RGB565

- CPP: 3
- BPC: 8 (actual: 5, 6)
- BPP: 3 × 8 = 24 bits
- TDATA width per pixel: ceil(24/8) × 8 = 24 bits

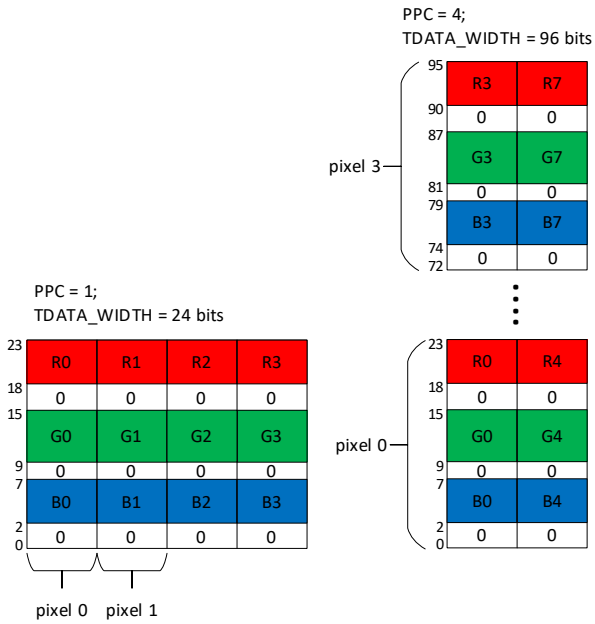

Figure 2.7. TDATA Pixel Mapping for Sample RGB565 Configuration

RGB888

- CPP: 3
- BPC: 8
- BPP: 3 × 8 = 24 bits
- TDATA width per pixel: ceil(24/8) × 8 = 24 bits

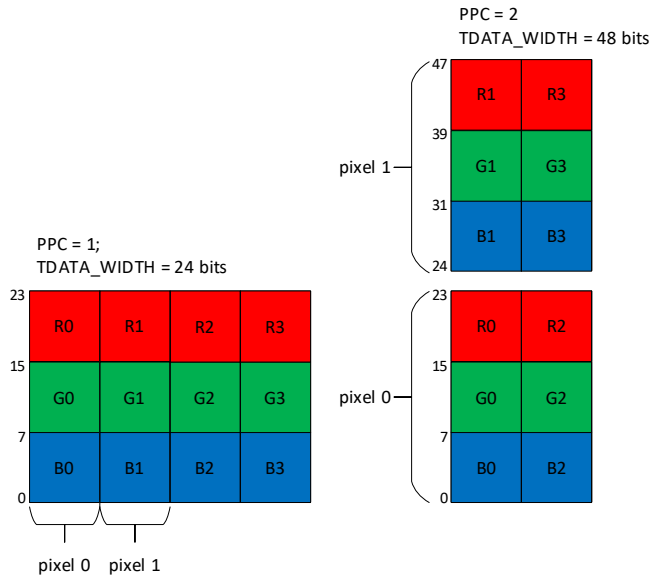

Figure 2.8. TDATA Pixel Mapping for Sample RGB888 Configuration

RAW10

- CPP: 1
- BPC: 10
- BPP: 1 × 10 = 10 bits
- TDATA width per pixel: ceil(10/8) × 8 = 16 bits

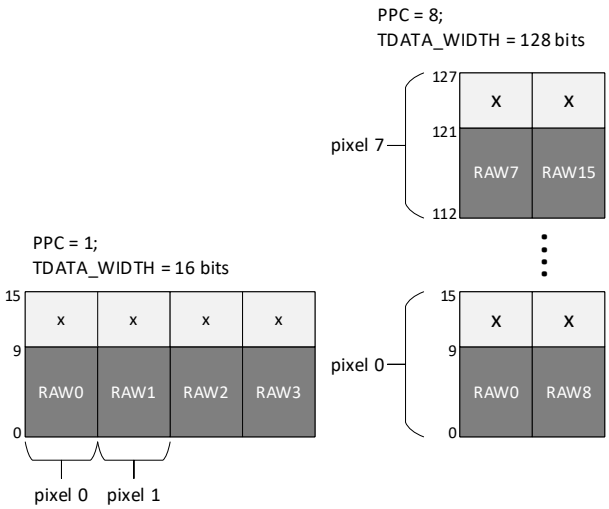

Figure 2.9. TDATA Pixel Mapping for Sample RAW10 Configuration

RAW12

- CPP: 1
- BPC: 12
- BPP: 1 × 12 = 12 bits
- TDATA width per pixel: ceil(12/8) × 8 = 16 bits

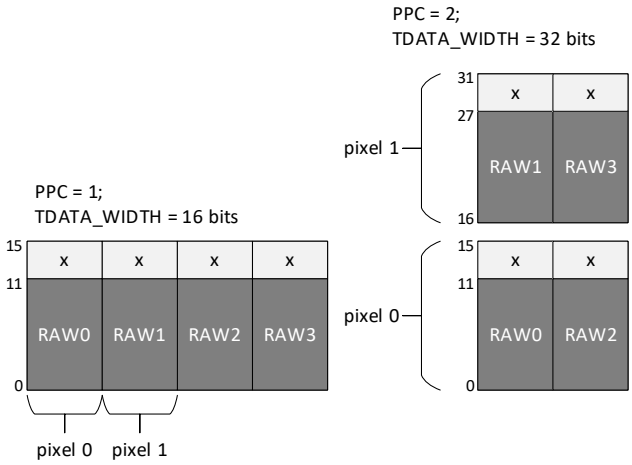

Figure 2.10. TDATA Pixel Mapping for Sample RAW12 Configuration

2.4.1.2. MIPI CSI/DSI Transmitter UVSI

Figure 2.11 illustrates the timing diagram for video frame transmission with blanking interval over UVSI during a TX transaction. In this integrated solution, the design includes a configurable line-buffer architecture, where the buffer depth can be configured according to the video line-packet size to support various resolutions and packet formats.

When sufficient horizontal blanking interval is provided between consecutive line packets on UVSI, the IP has sufficient time to complete the D-PHY LP-to-HS mode switching and to flush the stored data from the line buffer, enabling smooth and continuous operation without triggering backpressure.

When horizontal blanking is minimal or absent and line packets arrive back-to-back at UVSI, the IP must simultaneously store incoming data into the line buffer while requesting the D-PHY to transition from low power to high-speed mode. This LP to HS transition is not instantaneous and must meet the D-PHY timing parameters, such as tHS-PREPARE and tHS-ZERO, which introduce a finite delay. If additional lines continue to arrive during this transition and the buffer reaches the maximum capacity, the IP asserts backpressure on UVSI to prevent overflow and ensure reliable data transfer as shown in Figure 2.12.

You can increase the available buffering margin by adjusting the *Line Buffer Depth* parameter in the IP configuration. A larger *Line Buffer Depth* allows the IP to absorb consecutive line packets without prematurely asserting backpressure.

When backpressure is asserted (axis_vid_tready_o = low), the upstream video source is expected to pause pixel transmission immediately until tready returns high. The upstream logic must be able to throttle or halt data flow on tready deassertion, otherwise, incoming data may be lost or misaligned.

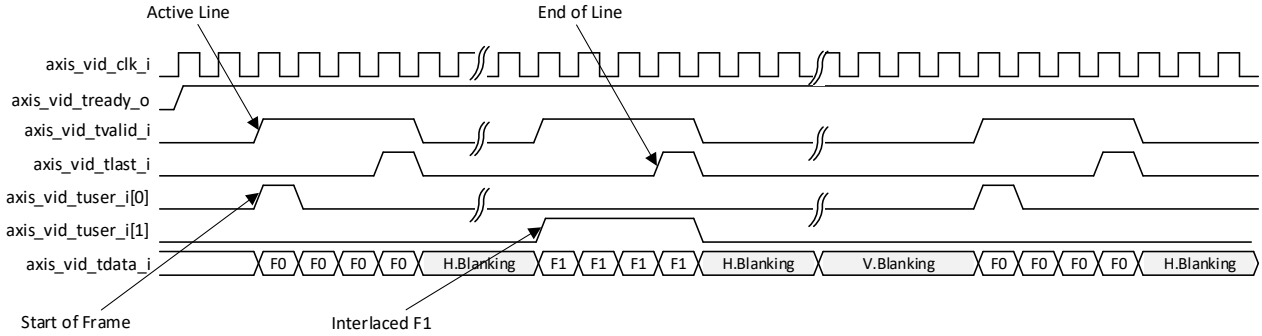

Figure 2.11. UVSI TX Timing Diagram with Blanking Interval

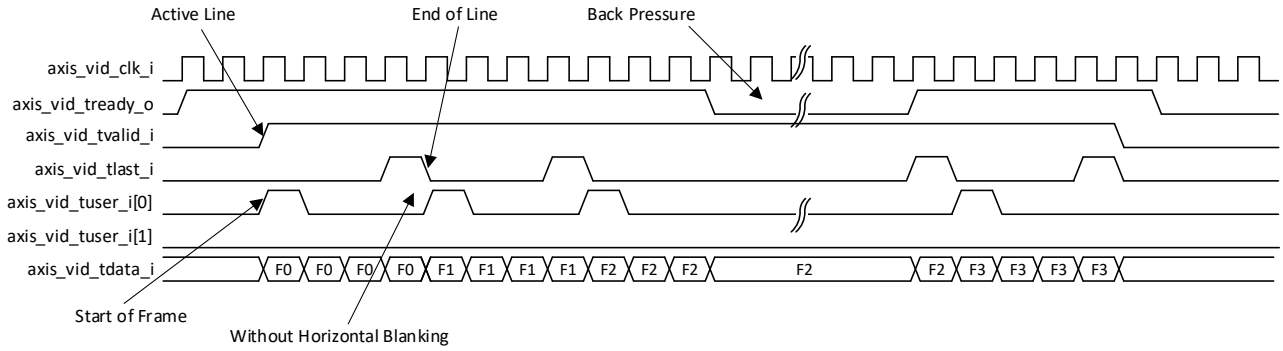

Figure 2.12. UVSI TX Timing Diagram without Blanking Interval

2.4.1.3. MIPI CSI/DSI Receiver UVSI

Figure 2.13 shows the timing diagram for video frame reception over UVSI where axis_vid_tready_i stays asserted during an RX transaction.

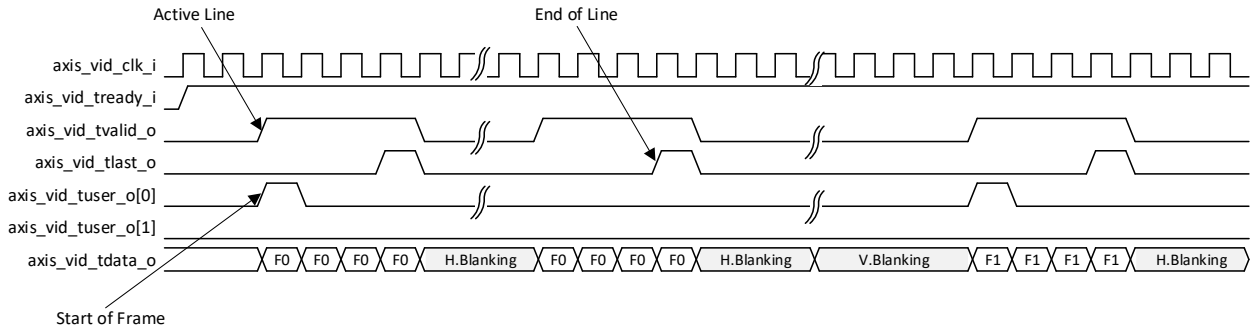

Figure 2.13. UVSI RX Timing Diagram without TREADY De-assertion

When axis_vid_tready_i is de-asserted (low) during an active packet transfer, as shown in Figure 2.14, pixels are stored in a pixel buffer. If axis_vid_tready_i stays low for a duration greater than or equal to the configured buffer depth (*Pixel Buffer Depth*), the buffer may overflow, which can cause functional errors.

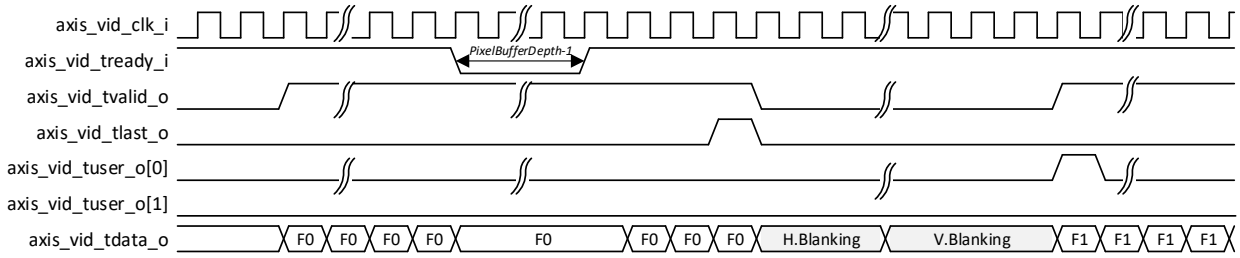

Figure 2.14. UVSI RX Timing Diagram with TREADY De-assertion

2.5. MIPI Timing Parameters

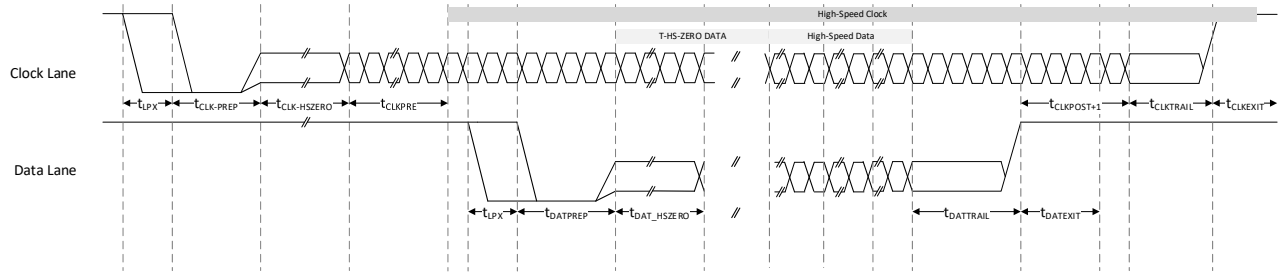

Figure 2.15. MIPI Timing Parameters

Figure 2.15 provides a visual representation of the MIPI D-PHY clock and data lane timing behavior, illustrating the relationship between signal transitions and the defined timing parameters. The timing parameters referenced in the MIPI CSI/DSI TX Protocol Timing Parameters section are depicted in the diagram using labeled intervals on the clock and data lanes. Each labeled interval corresponds to a specific timing parameter and indicates the location of delays and transition periods relative to the clock and data signals. Variations in timing parameter values are reflected as changes in the extent or placement of the corresponding waveform segments on the clock lane, the data lane, or both. When viewed together with the timing parameter definitions in the MIPI CSI/DSI TX Protocol Timing Parameters section, the diagram provides a clear mapping between the configured timing values and the resulting manifestation on the physical interface.

2.6. MIPI CSI/DSI Transmitter PHY Block

The PHY block (D-PHY transmitter) is the physical interface that drives MIPI D-PHY signals onto differential lanes and converts parallel byte data into serialized high-speed (HS) and low power (LP) signaling as required by the MIPI D-PHY specification. The PHY block supports the Soft D-PHY mode.

This block performs the following key functions:

- Clock generation: Produces the byte clock output from external high-speed clock and ensures proper distribution to the data lanes.
- Start-of-transmission (SoT) signaling: Generates the required HS prepare, HS-zero, and SoT sequences and appends the SYNC WORD pattern 0xB8 to incoming HS payload data before transmitting.
- End-of-transmission (EoT) signaling: Drives the required EoT and HS-trail patterns, followed by switching the lane back to LP mode.
- Serialization: Converts the incoming 8-bit parallel data into DDR-based serialized HS output on differential data lanes.
- Low power and high-speed control: Manages LP-to-HS and HS-to-LP transitions based on request inputs, ensuring compliance with MIPI D-PHY timing requirements.
- LP signaling: Drives LP-00 and LP-11 states for control sequences.

2.7. MIPI CSI/DSI Transmitter Controller Block

The TX controller block prepares outgoing video data and constructs MIPI-compatible packet headers and payloads before sending the data to the PHY for serialization. The controller block handles packet formatting, lane distribution, and all necessary packet-level control required for transmission.

This block performs the following key functions:

- Packetization: Forms MIPI packet headers and payloads from incoming byte-stream or pixel data, inserting Data Type, Word Count, Frame Number, Line Number, ECC, and CRC fields as required.
- Lane distribution: Splits the outgoing byte stream across multiple lanes based on the configured lane count, performing width expansion, byte mapping, and lane-based data scheduling.

2.8. MIPI CSI/DSI Transmitter Pixel-to-Byte Block

The Pixel-to-Byte module converts incoming pixel data into MIPI-compatible payload bytes and stores the data in the line buffer before forwarding the data to the packetization block through UVSI. This block bridges the video pixel clock domain (axis_vid_clk_i) and byte clock domain (byte_clk_o), using a dual-clock FIFO for buffering, synchronization, and flow-control management.

This block performs the following key functions:

- Pixel-to-Byte conversion: Converts incoming pixel data of various color formats into byte-aligned payload structures suitable for MIPI packetization. Implements AXI4-Stream TVALID/TREADY handshaking to manage data flow and prevent FIFO overflow or underflow during mode transitions or clock-rate differences.
- Line buffer: Stores pixel information in a byte-aligned format and performs clock-domain bridging between the video-pixel clock and the byte clock using a dual-clock FIFO, ensuring safe data transfer across asynchronous domains while managing backpressure.
- Video FSM: The video timing block defines and enforces the video cadence by sequencing synchronization events and active video transfer. This block drives packet generation of sync short packets (Frame Start, Frame End, Line Start, Line End) and active video packets. The video timing FSM operates in CSI-2 mode, selecting the protocol-specific packet schedule and timing rules. In accordance with the CSI-2 specification, the LS (Line Start) and LE (Line End) short packets are optional and may be disabled based on system configuration or video timing requirements using IP parameter.

Table 2.4. MIPI CSI/DSI Transmitter Pixel-to-Byte Supported Configurations

TX Controller Mode	Color Format	Number of Pixel per Clock (PPC)
32-bit	RGB888, RGB565, RAW10, RAW12	1, 2, 4

Table 2.5. MIPI CSI/DSI Transmitter Word Count Support Limitation

PHY Direction	PHY Mode	Protocol Mode	Controller Mode	Minimum Word Count Limitation (Bytes)
TX	Soft D-PHY	CSI-2	32-bit	32

2.9. MIPI CSI/DSI Transmitter Edge Clock Sharing

The Soft D-PHY TX uses FPGA DDR elements together with ECLK synchronizer and divider resources to generate the edge-aligned transmit clock. When *Edge Clock Mode* is External, the IP does not instantiate the ECLK synchronizer and divider chain. Instead, the ports required to drive the DDR path (such as the byte clock, ECLK sync clock, reset, and readiness indicators) are exposed as top-level inputs. These signals must be connected to the corresponding outputs of another D-PHY TX instance that has *Edge Clock Mode* set to Internal, instantiates the synchronizer and divider chain, and serves as the primary clock source. This shared-clock mode is intended for designs that instantiate multiple Soft D-PHY TX blocks within the same bank and need to conserve ECLK resources, provided all instances operate at the same bit rate.

Refer to the figure below for the expected connectivity.

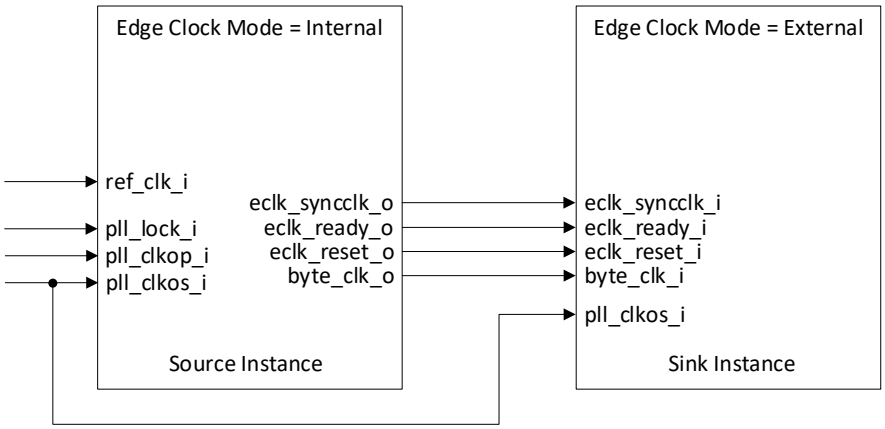

Figure 2.16. Edge Clock Sharing Port Connection

The source instance D-PHY TX supplies byte_clk_o, eclk_syncclk_o, eclk_reset_o, and eclk_ready_o to the sink instance D-PHY TX.

The sink instance connects the byte_clk_i, eclk_syncclk_i, eclk_reset_i, and eclk_ready_i ports directly to these source instance outputs, ensuring that both TX data paths remain synchronized to the same byte-clock domain.

This configuration allows multiple D-PHY transmitters to share one ECLK generation subsystem while maintaining deterministic DDR timing and aligned byte-clock behavior across all linked instances.

2.10. MIPI CSI/DSI Receiver PHY Block

The PHY block (D-PHY receiver) is the physical interface that receives MIPI D-PHY signals from differential pairs and converts them to parallel byte data. The PHY block supports the Soft D-PHY mode.

This block performs the following key functions:

- Clock recovery: Generates and distributes byte clock (byte_clk_o) from incoming MIPI clock lane.
- Deserialization: Uses DDR for 8-bit deserialization.
- Word alignment: Detects the 0xB8 SoTp and aligns incoming data.
- Low power and high-speed control: Manages LP-HS transitions.

2.11. MIPI CSI/DSI Receiver Controller Block

The RX controller block processes byte data from the PHY into MIPI packet headers and payloads. The block handles lane alignment, lane merging, and depacketization with error detection and correction.

This block performs the following key functions:

- Lane alignment: Aligns data across multiple lanes using FIFO-based buffering and SoTp detection.
- Lane merge: Combines aligned lane data into a single stream, performs width conversion, byte reordering, and propagates end-of-packet indicators.
- Depacketization: Extracts headers and payloads.

2.12. MIPI CSI/DSI Receiver Byte-to-Pixel Block

The Byte-to-Pixel module converts MIPI packet payload bytes into pixel data and outputs the data via UVSI AXI4-Stream. The block bridges the free-running clock domain (fr_clk_i) and video pixel clock domain (axis_vid_clk_i), and uses a dual clock FIFO for data buffering and flow control.

This block performs the following key functions:

- Byte-to-Pixel conversion: Translates MIPI CSI/DSI packet payload bytes into pixel data according to the configured video format. Payload processing begins after detecting SOF.
- Clock domain bridging: Interfaces between the free-running clock domain (fr_clk_i) and the video pixel clock domain (axis_vid_clk_i).
- AXI4-Stream output: Outputs the converted pixel data via UVSI.

Table 2.6. MIPI CSI/DSI Receiver Byte-to-Pixel Supported Configurations

RX Controller Mode	Color Format	Number of Pixel per Clock (PPC)
32-bit	RGB888, RGB565, RAW10, RAW12	1, 2, 4

Table 2.7. MIPI CSI/DSI Receiver Word Count Support Limitation

PHY Direction	PHY Mode	Protocol Mode	Controller Mode	Minimum Word Count Limitation (Bytes)
RX	Soft D-PHY	DSI-2	32-bit	14
RX	Soft D-PHY	CSI-2	32-bit	6

3. IP Parameter Description

The configurable attributes of the MIPI CSI/DSI IP are shown in the following tables. You can configure the IP by setting the attributes accordingly in the IP Catalog Module/IP Block Wizard of the Lattice Radiant software.

Wherever applicable, default values are in bold.

3.1. General

Table 3.1. General Attributes

Attribute	Selectable Values	Description
PHY Configuration		
PHY Direction	TX, **RX**	Selects the operating direction of the PHY.
PHY Mode	**Soft D-PHY**	Selects the physical layer implementation. Not editable. For information only.
PHY Data Width	**8**	Defines the width of the deserialized data stream per lane in bits. Not editable. For information only.
Number of D-PHY Lanes	1, 2, **4**	Specifies the number of data lanes. D-PHY supports up to 4 lanes (1-wire differential signaling per lane).
Line Rate Per Lane (Mbps)[1]	160–1500, **1000**	Maximum bandwidth when *PHY Direction* is *TX.*
	80–1500, **1000**	Maximum bandwidth when *PHY Direction* is *RX.* Set the data rate per lane in Mbps for D-PHY. This determines the maximum bandwidth capability of each lane.
Actual Byte Clock Frequency (MHz)	10–187.5, **125**	Operating byte clock frequency of the IP. The value is *Line Rate per Lane (Mbps)*/8. Not editable. For information only.
Actual D-PHY Clock Frequency (MHz)	40–750, **500**	Operating frequency of the PHY layer. The value is *Line Rate per Lane (Mbps)*/2. Not editable. For information only.
PHY Clock Mode	**Continuous**, Non-Continuous	Configures the clock mode for Soft D-PHY. Continuous mode keeps the clock lane in high-speed clock. Non-continuous mode allows the clock to enter low power state between high-speed data transmissions.
Edge Clock Mode	**Internal**, External	Selects the source of the edge clock (ECLK) used by the Soft D-PHY TX. ECLK is the high-speed per-bank clock network in Lattice FPGAs that drives DDR I/O elements and synchronizes PHY data transitions. Internal: The Soft D-PHY TX generates its own ECLK using internal ECLK synchronizer and divider resources. Use this mode when the D-PHY operates independently and can use dedicated per-bank ECLK resources. External: The Soft D-PHY TX receives ECLK from another D-PHY TX instance, enabling multiple TX instances in the same bank to share a common ECLK source. Use this mode to optimize clock resources and maintain synchronization across multiple TX instances running at the same bit rate. This setting is only available when *PHY Direction* is set to *TX.*
Reference Clock Frequency (MHz)	60–200, **100**	Sets the operating frequency of the clock domain for soft PHY related components in Soft D-PHY mode.
Controller		
Protocol Mode	DSI-2, **CSI-2**	Selects the MIPI protocol type. CSI-2 is optimized for camera and imaging applications with support for image data types and timing.

Attribute	Selectable Values	Description
		DSI-2 is designed for display applications with video-specific packet structures and flow control mechanisms. DSI-2 is only available when *PHY Direction* is set to *RX*.
Video Format		
Line Start and Line End	checked, **unchecked**	Checked: Activates LS and LE short-packet generation. When enabled, the video timing FSM issues Line Start and Line End short packet markers at each line boundary to indicate the beginning and end of active video lines. Unchecked: Disables LS and LE short-packet generation. The video timing FSM suppresses line-boundary markers and does not emit LS or LE short packets. This setting is only available when *PHY Direction* is set to *TX*.
Line Number Insertion	checked**, unchecked**	Checked: The module inserts the line number into the Line Start (LS) and Line End (LE) short packets. Unchecked: The line number field in both LS and LE packets is forced to 0. This setting is only available when *PHY Direction* is set to *TX* and *Line Start and Line End* is checked.
Number of Active Lines Per Frame	1–65535, **48**	Indicates how many active lines compose a video frame. The video timing FSM uses this value to track line progression and determine the frame boundary positions for generating Start of Frame (FS) and End of Frame (FE) short packets. This setting is only available when *PHY Direction* is set to *TX*.
Number of Pixel per Clock (PPC)	1, 2, **4**	Defines how many pixels are processed per clock cycle in the Pixel-to-Byte and Byte-to-Pixel conversion blocks. Higher PPC values increase throughput but require more parallel processing resources. The available options depend on the enabled color formats, PHY configuration, and timing requirements. Refer to the MIPI CSI/DSI Transmitter Pixel-to-Byte Block and MIPI CSI/DSI Receiver Byte-to-Pixel Block sections for more details.
Enable RGB888 Color Format	**checked**, unchecked	Enables support for RGB888 (24-bit) color format in the Pixel-to-Byte or Byte-to-Pixel conversion block.[2]
Enable RGB565 Color Format	checked, **unchecked**	Enables support for RGB565 (16-bit) color format in the Pixel-to-Byte or Byte-to-Pixel conversion block.[2]
Enable RAW10 Color Format	checked, **unchecked**	Enables support for RAW10 (10-bit) color format in the Pixel-to-Byte or Byte-to-Pixel conversion block.[2]
Enable RAW12 Color Format	checked, **unchecked**	Enables support for RAW12 (12-bit) color format in the Pixel-to-Byte or Byte-to-Pixel conversion block.[2]
Miscellaneous		
Line Buffer Depth	**depthN**, depthN×2, depthN×4	Defines the depth of the line-buffer FIFO used in the Pixel-to-Byte (P2B) block for TX. The FIFO has a fixed width of 256 bits. The required depth is determined by the number of active pixels per line and the selected bits-per-component. The minimum depth (depthN) is computed as the smallest power-of-two FIFO depth that can hold one full video line, using ceil() to ensure the FIFO is never undersized. You may select depthN×2 or depthN×4 to increase buffering margin when horizontal blanking is short or absent. This attribute is only available when *PHY Direction* is set to *TX*. $\mathrm{FIFO}_{\mathrm{DEPTH}} = \mathrm{Ceil}((\mathrm{Hactive} \times \mathrm{BPC})/256)$ $\mathrm{dephtN} = 2^{ceil[log_2(FIFO\_DEPTH)]}$

Attribute	Selectable Values	Description
		Where Hactive: Number of Active Pixels Per Line BPC: Bits Per Component
Pixel Buffer Depth	**depthN×1**, depthN×2, depthN×4, depthN×8	Specifies the depth of the internal pixel data buffer on Byte-to-Pixel (B2P) block of RX. The default min setting, depthN, is optimized to use one EBR. Nexus EBR size = 18432 bits $\text{dephtN} = 2^{floor\left[log_2\left(\frac{EBR\ size\ (bits)}{TDATA\_WIDTH+2}\right)\right]}$ Example 1: TDATA_WIDTH = 96 depthN = 2^floor[log2(18432/(96+2))] depthN = 128 GUI Options: 128, 256, 512, 1024 Example 2: TDATA_WIDTH = 64 depthN = 2^floor[log2(18432/(64+2))] depthN = 256 GUI Options: 256, 512, 1024, 2048 This setting is only available when *PHY Direction* is set to *RX*.
Maximum Buffered Pixels	For *PHY Direction* == TX, [floor((*line_buffer_depth* / bits_per_pixel) × 256] For *PHY Direction* == RX, [*Pixel Buffer Depth* × *Number of Pixel per Clock (PPC)*]	Indicates the total number of pixels that can be buffered in the FIFO based on *Line Buffer Depth* and selected pixel format when *PHY Direction* is set to TX or *Pixel Buffer Depth* when *PHY Direction* is set to RX. Not editable. For information only.
Enable tINIT Counter	checked, **unchecked**	Enables the internal T_INIT counter that ensures the D-PHY TX remains in the LP-11 Stop State for the required initialization period before entering high-speed transmission. When disabled, the TX skips the enforced T_INIT wait time. This setting is only available when *PHY Direction* is set to *TX*.
tINIT Counter Value	1–65535, **1000**	Specifies the number of byte clock cycles the transmitter must remain in the LP-11 Stop State to meet the D-PHY T_INIT initialization requirement. This setting is only available when *PHY Direction* is set to *TX* and *Enable tINIT Counter* is checked.
tINIT Duration (ns)	**5333.33**	Displays the effective tINIT duration in ns computed from the user-programmed *tINIT counter value*. tINIT_ns = *tINIT counter value* × Byte_Clock_Period_ns This setting is only available when *PHY Direction* is set to *TX*. Not editable. For information only.

Notes:

1. The maximum data rate depends on the family, package, and performance grade of the device. Check the device data sheet for more information.
2. Only one-color format can be enabled at a time.

3.2. MIPI CSI/DSI TX Protocol Timing Parameters

Table 3.2. General Attributes

Attribute	Selectable Values	Description
Protocol Timing Parameters		
TX Global Operation Timing Parameters		
t_LPX	1–255[4]	Duration of any low-power state.
t_HS-PREPARE	1–255	Duration of the LP-00 line state before the HS-0 line state. Default value of *t_HS-PREPARE = ceil*(Minimum timing parameter value/ByteClk Period) If *t_HS-PREPARE_val* > 0, the default value of *t_HS-PREPARE* is *t_HS-PREPARE_val* + 1, otherwise, the default value is 1.
t_HS-ZERO	1–255	Delay from LP 00 entry to the point when the IP is ready to begin HS transmission. The actual HS-ZERO on the D-PHY data lanes depends on the serializer delay and internal packet processing. Default value of *t_HS_ZERO* = ceil(Minimum timing parameter value/ByteClk Period).
t_HS-TRAIL	3–255	Duration of the flipped bit after the last payload data bit of an HS transmission burst. Default value of *t_HS_TRAIL* = floor(Maximum T_EOT timing parameter value/ByteClk Period).
t_HS-EXIT	1–255	Duration of the data LP-11 state following an HS transmission burst. Default value of *t_HS_EXIT* = floor((Minimum timing parameter value/ByteClk Period)+1).
t_CLK-PREPARE	1–255	Duration of the LP-00 clock state immediately before the HS-0 clock state in the LP-to-HS sequence. Default value of *t_CLK-PREPARE* = ceil(Minimum timing parameter value/ByteClk Period).
t_CLK-ZERO	1–255	Duration of the clock HS-0 state prior to starting the toggling of the high-speed clock. When *CIL Bypass* == Checked: Default value of *t_CLK-ZERO* = ceil(Minimum timing parameter value/ByteClk Period).
t_CLK-PRE	2–255	Duration of the HS clock prior to the start of the LP-to-HS sequence of the data lanes. Default value of *t_CLK-PRE* = ceil((Minimum timing parameter value/ ByteClk Period)+1).
t_CLK-POST	2–255[4]	Duration of the HS clock after the last associated data lane has transitioned to LP mode. The interval is defined as the period from the end of tHS-TRAIL to the beginning of tCLK-TRAIL.
t_CLK-TRAIL	2–255	Duration of the HS-0 state after the last clock bit of an HS transmission burst. When *CIL Bypass* == Checked: Default value of *t_CLK-TRAIL* = floor((Minimum timing parameter value/ByteClk Period) + 2).
t_CLK-EXIT	1–255[4]	Duration of the clock LP-11 state following an HS transmission burst to the next HS transmission burst in non-continuous clock mode.

Notes:

1. The maximum data rate depends on the family, package, and performance grade of the device. Check the device data sheet for more information.
2. Only one-color format can be enabled at a time.
3. The duration of the timing parameter is equal to (byte clock period) × (attribute value).
4. The default value of this parameter is equal to floor((Minimum timing parameter value/ByteClk Period) + 1).

5. The timing parameters are in number of byte clock cycles. The IP automatically calculates the timing parameters (in byte-clock cycles) to meet the required timing limits. You must verify that the selected values meet your overall system timing, as actual D-PHY lane timing may vary because of internal serialization and register delays.

4. Signal Description

This section describes the MIPI CSI/DSI IP ports.

4.1. MIPI CSI/DSI TX Signal Description

4.1.1. Clock Interface

Table 4.1. Clock and Reset Ports

Port	Type	Description
ref_clk_i	Input	Clock input for soft PHY fabric-based implementation. This clock drives all soft PHY related logic and follows the frequency set in the *Reference Clock Frequency (MHz)* attribute. Available if *PHY Mode* == *Soft D-PHY* and *Edge Clock Mode* == *Internal*.
ref_rst_n_i	Input	Asynchronous active-low reset input for soft D-PHY. This is synchronized internally with ref_clk_i clock. This resets all logic and blocks in the ref_clk_i domain. Available if *PHY Mode* == *Soft D-PHY* and *Edge Clock Mode* == *Internal*.
pll_clkop_i	Input	External PLL clock input operating at the D-PHY high-speed clock frequency. This is the 0-degree phase PLL output used as the primary reference for generating the HS clock lane. Available if *PHY Mode* == *Soft D-PHY* and *Edge Clock Mode* == *Internal*.
pll_clkos_i	Input	External PLL clock input that is 90 degrees phase-shifted from pll_clkop_i. Provides the quadrature HS clock required for DDR edge alignment in the D-PHY clock lane. Available if *PHY Mode == Soft D-PHY*. Refer to the MIPI CSI/DSI Transmitter Edge Clock Sharing section for more details.
axis_vid_clk_i	Input	The video-clock frequency is normally determined by the selected video parameters such as resolution, frame rate, and pixel format. The IP includes a configurable line-buffer that operates using a store-and-forward mechanism. This mechanism allows the transmitter to run at a pixel-clock frequency lower than the raw throughput implied by the video format, provided the selected *Line Buffer Depth* is sufficient to absorb the incoming line data without causing backpressure.
axi_vid_rst_n_i	Input	Active-low AXI4-Stream video reset input for the Pixel-to-Byte converter. This is synchronized internally with axis_vid_clk_i. Resets logic in pixel domain.
Edge Clock Sharing		
byte_clk_i	Input	Byte clock input. Connects to byte_clk_o of source instance with *Edge Clock Mode == Internal*. Available if *Edge Clock Mode == External*[3].
eclk_syncclk_i	Input	Drives the ECLK pin of DDR modules of data path. Connects to eclk_syncclk_o of source instance with *Edge Clock Mode == Internal*. Available if *Edge Clock Mode == External*[3].
eclk_reset_i	Input	Drives the reset port of DDR modules. Connects to eclk_reset_o of source instance with *Edge Clock Mode == Internal*. Available if Edge Clock Mode == *External*[3].
eclk_ready_i	Input	Indicates the clock synchronization status to sink instance. Connects to eclk_ready_o of source instance with *Edge Clock Mode == Internal*. Available if *Edge Clock Mode == External*[3].

Port	Type	Description
byte_clk_o	Output	Byte clock output. Byte Clock is generated by gearing down pll_clkop_i and operates as a divided version of the PLL output clock. Connects to byte_clk_i of sink instance with *Edge Clock Mode == External.* Available if *Edge Clock Mode == Internal*[3].
eclk_syncclk_o	Output	Drives the ECLK pin of DDR modules of data path. Connects to eclk_syncclk_i of sink instance with *Edge Clock Mode == External.* Available if *Edge Clock Mode == Internal*[3].
eclk_reset_o	Output	Drives the reset port of DDR modules. Connects to eclk_reset_i of sink instance with *Edge Clock Mode == External*. Available if *Edge Clock Mode == Internal*[3].
eclk_ready_o	Output	Indicates the clock synchronization status of source instance. Connects to eclk_ready_o of sink instance with *Edge Clock Mode == External.* Available if *Edge Clock Mode == Internal*[3].

Notes:

1. Refer to the Unified Video Streaming Interface (UVSI) section for details.
2. The maximum supported clock frequency depends on the family, package, and performance grade of the device. Check the device data sheet for more information.
3. Refer to the MIPI CSI/DSI Transmitter Edge Clock Sharing section for more details.

4.1.2. D-PHY Interface

Table 4.2. D-PHY Interface Ports

Port	Type	Description
clk_p_io	Inout	Positive differential clock I/O pad for D-PHY implementations.
clk_n_io	Inout	Negative differential clock I/O pad for D-PHY implementations.
d_p_io[NUMLANE[1]-1:0]	Inout	Positive differential data I/O pad for D-PHY implementations.
d_n_io[NUMLANE[1]-1:0]	Inout	Negative differential data I/O pad for D-PHY implementations.

Note:

1. Corresponds to the *Number of D-PHY Lanes* attribute.

4.1.3. Unified Video Streaming Interface

Table 4.3. Unified Video Streaming Interface Ports

Port	Type	Description
axis_vid_tready_o	Output	AXI4-Stream ready signal from the TX module. Indicates that the module is ready to accept new pixel data. When this signal is low, upstream video logic must pause data transmission.
axis_vid_tvalid_i	Input	AXI4-Stream valid signal from upstream video logic. Indicates that axis_vid_tdata_i contains valid pixel data for transmission.
axis_vid_tdata_i[TDATA_WIDTH -1:0]	Input	AXI4-Stream pixel data bus carrying incoming formatted pixel data. Pixel values are packed LSB-first. Multiple pixels may be concatenated when PPC > 1. Data is accepted only when axis_vid_tvalid_i and axis_vid_tready_o are both high. Refer to the TDATA Mapping and Unified Video Streaming Interface (UVSI) sections for details.
axis_vid_tuser_i[1:0]	Input	AXI4-Stream video user sideband signals. Provides frame and line timing information. Bit [0]: Start of Frame (SOF) – Asserts for first pixel of each frame. Bit [1]: Interlaced F1 – Currently unused and is tied to low.
axis_vid_tlast_i	Input	AXI4-Stream End-of-Line indicator. Asserts with the last pixel of each video line and marks completion of a line transfer.

4.1.4. Miscellaneous

Table 4.4. Miscellaneous Ports

Port	Type	Description
tinit_done_o	Output	Indicates completion of the D-PHY LP-11 initialization period. When *Enable tINIT Counter* is checked, this signal asserts after (*tINIT counter value* – 1) byte-clock cycles, representing the programmed LP-11 duration. When the *Enable tINIT Counter* is unchecked, the signal asserts automatically when the internal soft PHY has entered and stabilized in LP-11 state, indicating that the link is ready for HS entry.

4.2. MIPI CSI/DSI RX Signal Description

4.2.1. Clock Interface

Table 4.5. Clock and Reset Ports[1]

Port	Type	Description
ref_clk_i	Input	Clock input for soft PHY fabric-based implementation. This clock drives all soft PHY related logic. Available if *PHY Mode == Soft D-PHY*.
ref_rst_n_i	Input	Asynchronous active-low reset input for soft D-PHY. This is synchronized internally with ref_clk_i clock. This resets all logic and blocks in the ref_clk_i domain. Available if *PHY Mode == Soft D-PHY*.
byte_clk_o	Output	Byte clock output recovered from the MIPI data stream. Represents the rate at which payload bytes are extracted from received packets. This is only active when the clock lane is in high-speed mode. Frequency is displayed as *Actual Byte Clock Frequency (MHz).*
fr_clk_i	Input	Free-running clock. Determines the rate at which received MIPI packets are processed. Drives the logic that detects the low power to high-speed transition of clock and data lanes.
fr_rst_n_i	Input	Asynchronous active-low reset input. This is synchronized internally with fr_clk_i clock. Resets all logic and blocks in the fr_clk_i domain.
axis_vid_clk_i	Input	AXI4-Stream video clock input for the Byte-to-Pixel converter output interface. Drives the pixel processing pipeline including format conversion, pixel packing, and AXI4-Stream protocol generation.
axis_vid_rst_n_i	Input	Active-low AXI4-Stream video reset input for the Byte-to-Pixel converter. This is synchronized internally with axis_vid_clk_i. Resets logic in pixel domain.

Note:

1. The operating clock frequency and requirements information are available in the MIPI CSI/DSI Receiver Clocking Overview section.

4.2.2. D-PHY Interface

Table 4.6. D-PHY Interface Ports

Port	Type	Description
clk_p_io	Inout	Positive differential clock I/O pad for D-PHY implementations.
clk_n_io	Inout	Negative differential clock I/O pad for D-PHY implementations.
d_p_io[NUMLANE[1]-1:0]	Inout	Positive differential data I/O pad for D-PHY implementations.
d_n_io[NUMLANE[1]-1:0]	Inout	Negative differential data I/O pad for D-PHY implementations.

Note:

1. Corresponds to the *Number of D-PHY Lanes* attribute.

4.2.3. Unified Video Streaming Interface

Table 4.7. Unified Video Streaming Interface Ports

Port	Type	Description
axis_vid_tready_i	Input	AXI4-Stream video ready input from downstream video processing logic. When low, causes the byte-to-pixel converter to pause pixel output and may trigger internal buffering or flow control mechanisms.
axis_vid_tvalid_o	Output	AXI4-Stream video valid output. Indicates availability of valid pixel data on axis_vid_tdata_o. Used for standard AXI4-Stream handshaking protocol.
axis_vid_tdata_o[TDATA_WIDTH-1:0]	Output	AXI4-Stream video data output containing converted pixel data in the selected format. Pixel data is packed LSB-first. Multiple pixels are concatenated for *Number of Pixel per Clock (PPC)* > 1. Valid only when axis_vid_tvalid_o is asserted. Minimum TDATA_WIDTH width is 1 byte and may include padding bits for alignment to byte boundaries. Refer to the Unified Video Streaming Interface (UVSI) section for details.
axis_vid_tuser_o[1:0]	Output	AXI4-Stream video user sideband signals. Provides frame and line timing information. Bit [0]: Start of Frame (SOF) – Asserts for first pixel of each frame. Bit [1]: Interlaced F1 – Currently unused and is tied to low.
axis_vid_tlast_o	Output	AXI4-Stream video signal indicating the last pixel transfer of a video line. This signal always coincides with the last pixel of the line.

5. Designing with the IP

This section provides information on how to generate the IP core using the Lattice Radiant software and how to run simulation and synthesis. For more details on the Lattice Radiant software, refer to the Lattice Radiant software user guide.

Note: The screenshots provided are for reference only. Details may vary depending on the version of the IP or software being used. If there have been no significant changes to the GUI, a screenshot may reflect an earlier version of the IP.

5.1. Generating and instantiating the IP

You can use Lattice Radiant software to generate IP modules and integrate them into the device architecture. To generate the MIPI CSI/DSI IP in the Lattice Radiant software, follow these steps:

1. Create a new Lattice Radiant software project or open an existing project.
2. Click the **IP Catalog** button to view the **IP Catalog** pane.
3. On the **IP on Local** tab, double-click **MIPI CSI/DSI** under the **IP, Audio_Video_and_Image_Processing** category. The **Module/IP Block Wizard** opens.
   Note: If the IP is not available on the **IP on Local** tab, download the IP from the **IP on Server** tab.

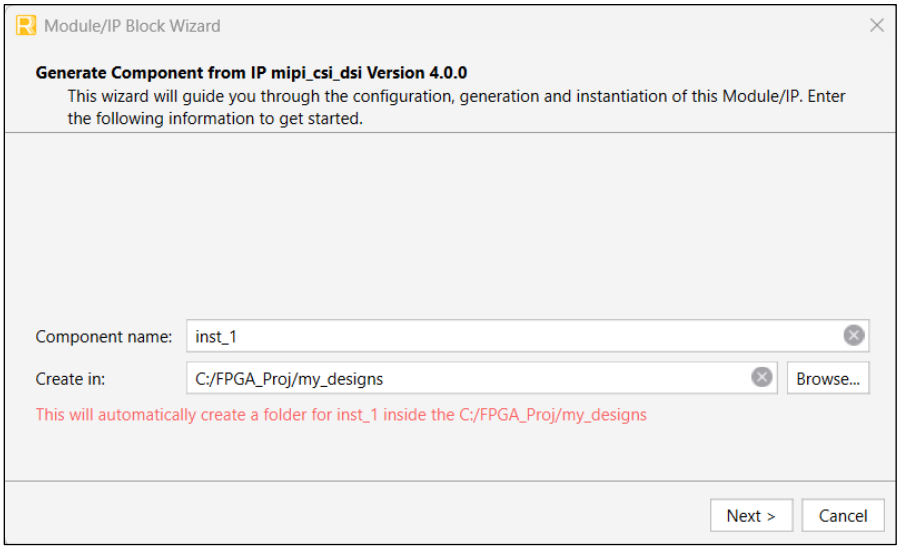

Figure 5.1. Module/IP Block Wizard

4. Enter values in the **Component name** and **Create in** fields, then click **Next**.
5. Customize the selected MIPI CSI/DSI IP using drop-down lists and check boxes. The figures below show the example configurations of the MIPI CSI/DSI IP. For details on the configuration options, refer to the IP Parameter Description section.

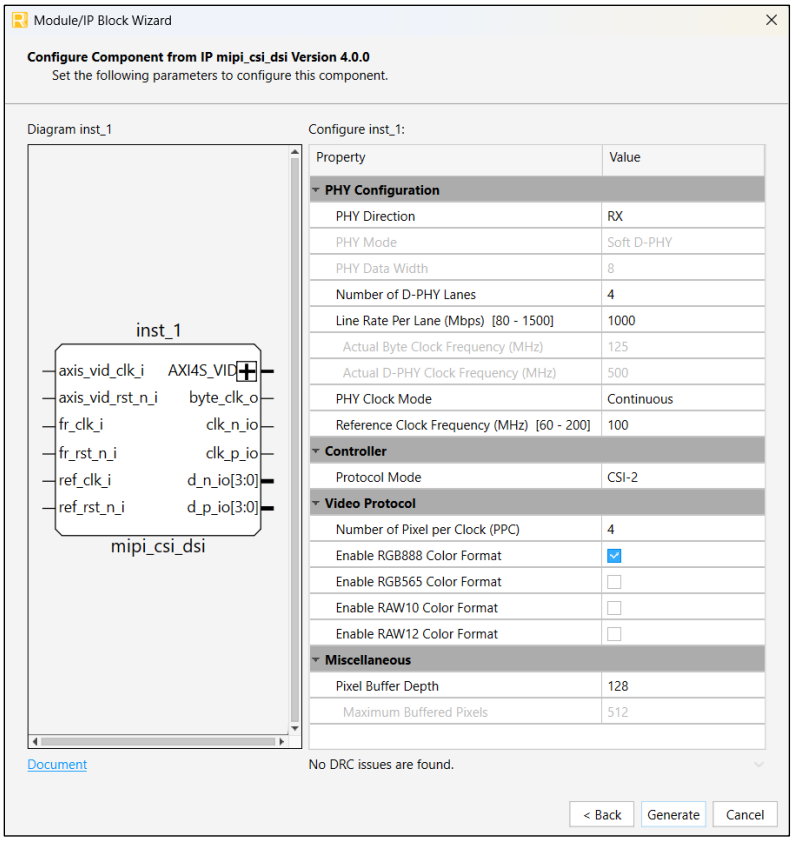

Figure 5.2. MIPI CSI/DSI IP Receiver Mode Configuration

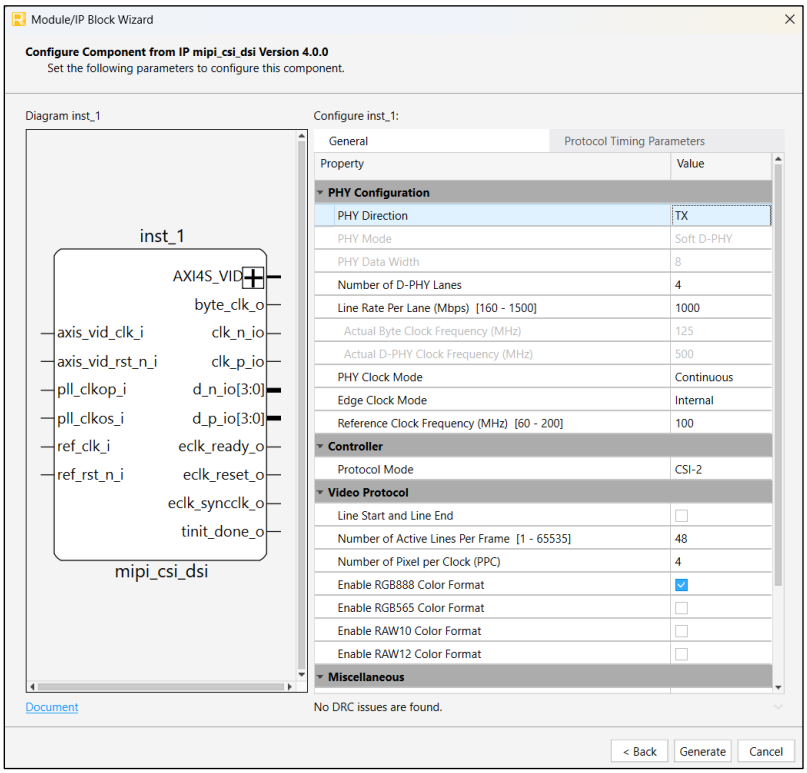

Figure 5.3. MIPI CSI/DSI IP Transmitter Mode Configuration

6. Click **Generate**. The **Check Generated Result** window opens. This window shows design block messages and results.

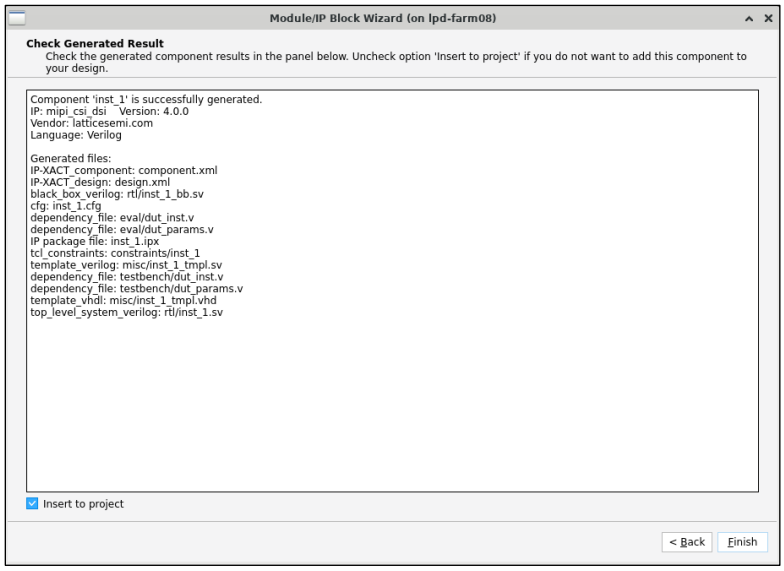

Figure 5.4. Check Generated Result

7. Click **Finish**. All generated files are placed in the directory specified by the **Component name** and **Create in** fields shown in Figure 5.1.

5.1.1. Generated Files and File Structure

The generated MIPI CSI/DSI module package includes the closed-box (*<Component name>_bb.v*) and instance templates (*<Component name>_tmpl.v/vhd*) that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file (*<Component name>.v*) that can be used as an instantiation template for the module is also provided. You may also use this example as the starting template for your top-level design. The generated files are listed in Table 5.1.

Table 5.1. Generated File List

Attribute	Description
<Component name>.ipx	Contains the information on the files associated with the generated IP.
<Component name>.cfg	Contains the parameter values used in IP configuration.
component.xml	Contains the ipxact: component information of the IP.
design.xml	Documents the configuration parameters of the IP in the IP-XACT 2014 format.
rtl/<Component name>.v	Provides an example RTL top file that instantiates the module.
rtl/<Component name>_bb.v	Provides the synthesis closed-box.
misc/<Component name>_tmpl.v misc /<Component name>_tmpl.vhd	Provides instance templates for the module.

5.2. Design Implementation

Completing your design includes additional steps to specify analog properties, pin assignments, and timing and physical constraints. You can add and edit the constraints using the Device Constraint Editor or by manually creating a .pdc file.

Post-Synthesis constraint files (.pdc) contain both timing and non-timing *constraint.pdc* source files for storing logical timing/physical constraints. Constraints that are added using the Device Constraint Editor are saved to the active .pdc file. The active post-synthesis design constraint file is then used as input for post-synthesis processes.

Refer to the relevant sections in the Lattice Radiant software user guide for more information on how to create or edit constraints and how to use the Device Constraint Editor.

5.3. Timing Constraints

The MIPI CSI/DSI IP generates the following constraint files:

- A constraint file in SDC format (*<ip_instance_path>/constraints/constraint.sdc*) that contains both pre-synthesis and post-synthesis IP constraints. These constraints are automatically used and propagated by the software tool starting from the Lattice Radiant software version 2024.1. These constraints can be modified if you have a thorough understanding of the effect of each constraint.
- An evaluation post-synthesis constraint file in PDC format (*<ip_instance_path>/eval/constraint.pdc*).

To run the software implementation flow using the provided evaluation file after the IP is generated, follow these steps:

1. In the **Post-Synthesis Constraint Files** section, add *<ip_instance_path>/eval/constraint.pdc*.
2. Run the implementation flow.

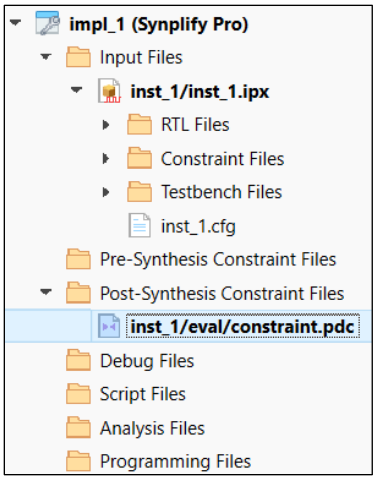

Figure 5.5. Example Evaluation Project Settings

Notes:

- You need to provide proper timing and physical design constraints to ensure that your design meets the desired performance goals for the FPGA.
- The constraint files have been verified during IP evaluation with the evaluation wrapper instantiated directly in the top-level module.
- During synthesis, you can ignore clock related warnings as the evaluation IP does not include clock-related constraints at pre-synthesis level.
- During post-synthesis, there may be warnings related to dropped constraints. As the IP supports many configurations and parameter combinations, some default constraints may not be applicable to the selected configuration.
- In transmitter mode, if *Edge Clock Mode* is set to Internal, you may encounter the Place and Route error as the edge clock synchronizer clock (eclk_syncclk_o) is illegally mapped to the FPGA I/O. This clock is intended to be connected to a DDR primitive or just left unconnected if unused.

For more information on timing constraints, refer to the Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059).

5.4. Physical Constraints

For MIPI CSI/DSI receiver Soft D-PHY mode, if I/O top-level ports are not automatically routed to MIPI I/O, define the MIPI pins (clk_p_io, d_p_io_*) with IO_TYPE of MIPI_DPHY in the .pdc file. An example is shown in Figure 5.6.

```
ldc_set_port -iobuf {IO_TYPE=MIPI_DPHY} [get_ports {clk_p_io*}]
ldc_set_port -iobuf {IO_TYPE=MIPI_DPHY} [get_ports {d_p_io*}]
```

Figure 5.6. Defining MIPI D-PHY Port Pins Using MIPI_DPHY IO_TYPE

5.5. Specifying the Strategy

The Lattice Radiant software provides two predefined strategies: Area and Timing. The software also enables you to create customized strategies. For details on how to create a new strategy, refer to the Strategies section of the Lattice Radiant software user guide.

5.6. Running Functional Simulation

You can run functional simulation after the IP is generated.

To run functional simulation, follow these steps:

1. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in Figure 5.7.

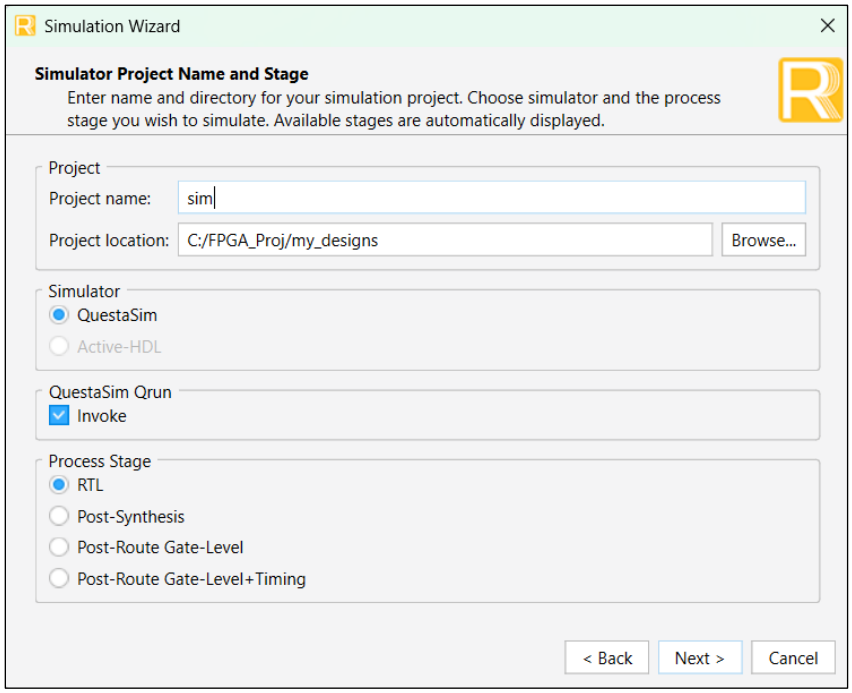

Figure 5.7. Simulation Wizard

2. Click **Next** to open the **Add and Reorder Source** window as shown in Figure 5.8.

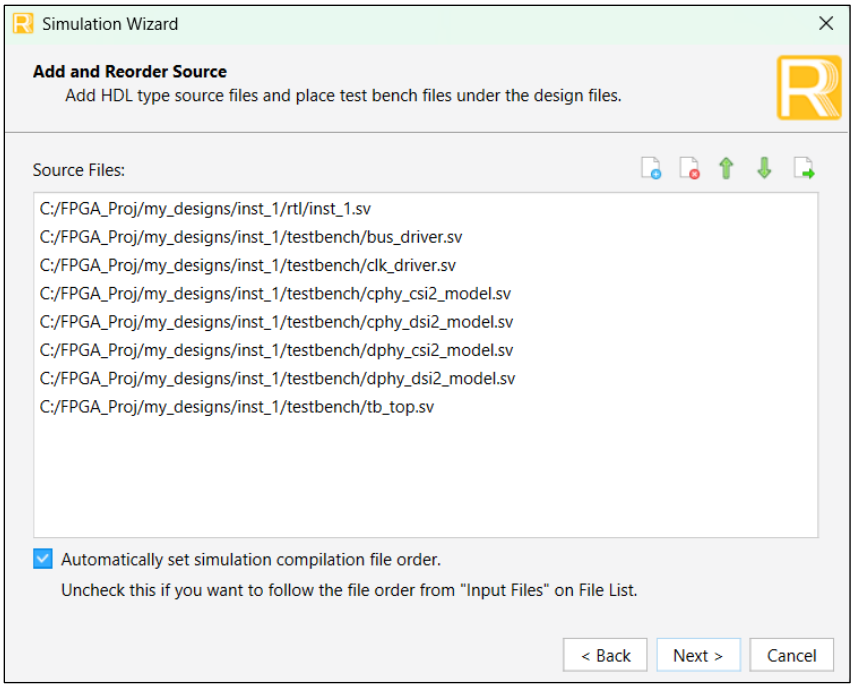

Figure 5.8. Add and Reorder Source

3. Click **Next**. The **Summary** window is shown.
4. Click **Finish** to run the simulation.

The waveform in Figure 5.9 shows an example simulation result.

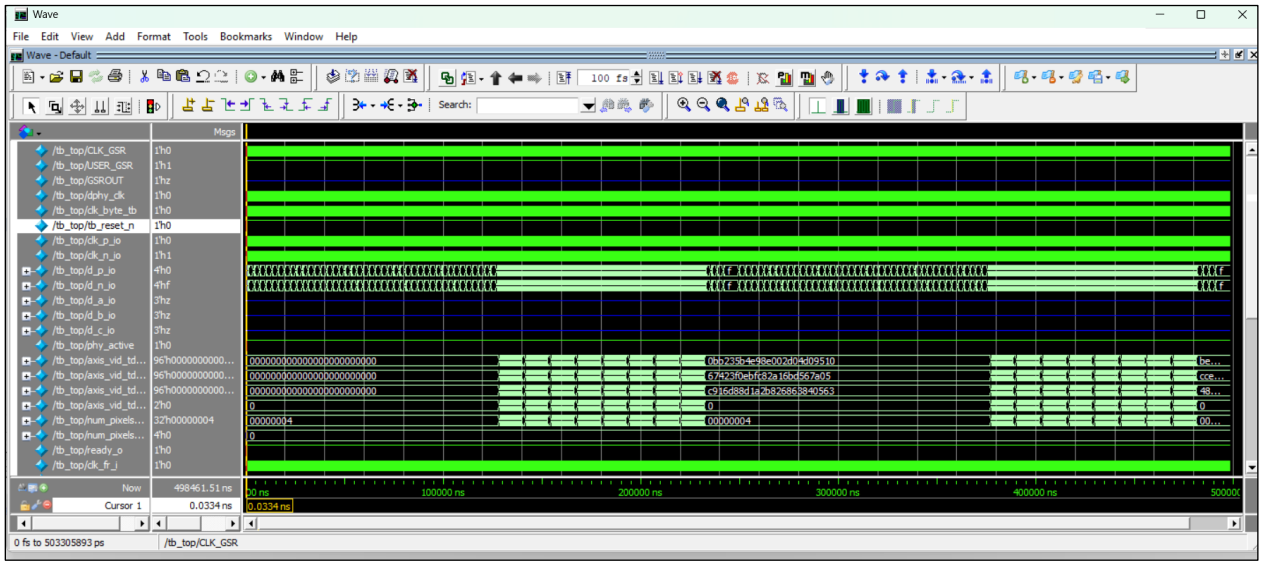

Figure 5.9. Simulation Waveform

5.6.1. Simulation Results

When the simulation is complete, the output in the Transcript window is shown in the following figure.

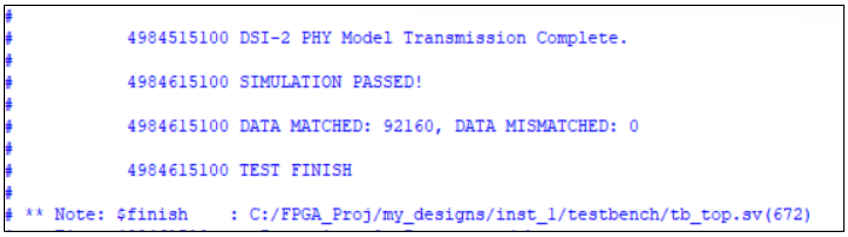

Figure 5.10. Simulation Result

6. Debugging

This section lists possible issues and suggested troubleshooting steps that you can follow.

6.1. Debug Methods

6.1.1. PHY Clock Status in Transmitter Mode

In transmitter mode, tinit_done_o is provided as an output status indicator to help you observe when the PHY has completed the initialization sequence and internal PHY reference clock has synchronized.

This allows you to ensure that the PHY is fully initialized and ready for normal operation before starting any LP or HS transactions. For additional information, refer to Table 4.4.

6.1.2. Generated byte_clk_o Does Not Toggle in Receiver Mode

In non-continuous PHY clock mode, byte_clk_o toggles only during high-speed data transmission on the MIPI D-PHY lanes. In continuous PHY clock mode, byte_clk_o toggles all the time.

If this clock does not toggle as expected:

- Ensure the upstream source is active. For example, camera sensor or an application processor.
- Ensure the upstream device configuration matches the IP settings, such as PHY clock mode and bit rate.
- Verify that all input clocks and reset signals to the IP module are correctly set up as recommended.

6.1.3. Data Appears Corrupted in Receiver Mode

If the IP output data appears corrupted, check whether axis_vid_tready_i goes low during active packet transmission. If this signal remains low for more than (*Pixel Buffer Depth-1)* axis_vid_clk_i cycles, the internal buffer may overflow, leading to functional issues. To prevent overflow, ensure axis_vid_tready_i does not stay low beyond the allowed limit. For details, refer to the MIPI CSI/DSI Receiver UVSI section.

6.1.4. Missing Packets in Receiver Mode

If there are any missing packets in the IP output data, check if all input clock frequencies and $t_{HS\text{-}PREPARE}$ + $t_{HS\text{-}ZERO}$ timing requirements specified in the Limitations for MIPI CSI/DSI Receiver section are met.

6.2. Debug Tools

You can use various tools to debug MIPI CSI/DSI IP design issues.

6.2.1. 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 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 set 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.

7. Design Considerations

7.1. Limitations for MIPI CSI/DSI Transmitter

- Escape Mode, Ultra Low Power State (ULPS), and Bus Turnaround sequences are not yet supported.
- Minimum word count supported is dependent on IP setting. Refer to Table 2.3 for details.
- The IP supports only video resolutions where the CSI-2 Word Count (WC) per active line is 32-byte aligned. You can verify the video resolution support using the following condition:

  $$WC \,\%\, 32 = 0$$

  Where WC is the number of payload bytes for one active line, % is a modulo operator based on the selected pixel format. For example, WC = 3 × Hactive for RGB888, WC = 2 × Hactive for RGB565.
  - If the computed WC value is not divisible by 32, the resolution is not supported.
  - The IP uses the % symbol as the modulo operator (remainder operation).
- UVSI interlaced format is not supported.
- Trail duration may exceed maximum T_{EOT} because of limitations in design implementation when *Number of TX Lanes* > 1 for the following configurations:
  - *D-PHY TX IP* is Soft D-PHY. For this configuration, consider reducing the trail timing parameter by 1.
  - After reducing the trail timing parameter by 1, timing violations may still be observed at the lowest data rates because of the wider byte clock period, which further constrains the trail timing margin.
- Some IP configurations may have slower Fmax when used in devices with slow performance grade.

7.2. Limitations for MIPI CSI/DSI Receiver

- Escape Mode, Ultra Low Power State (ULPS), and Bus Turnaround sequences are not yet supported.
- Error detection is not supported.
- Minimum word count supported is dependent on IP setting. Refer to Table 2.5 for details.
- UVSI interlaced format is not supported.
- When *PHY Mode* == Soft D-PHY, the minimum duration of $t_{HS\text{-}PREPARE}$ + $t_{HS\text{-}ZERO}$ from D-PHY source must meet the following condition:

  $$t_{HS-PREPARE} + t_{HS-ZERO} \geq 145\,ns + 10\ \text{UI} + (8 \times t_{CLK\_SYNC})$$

  where t_{CLK_SYNC} is the period of ref_clk_i.
- When *PHY Mode* == Soft D-PHY and the corresponding byte clock is ≤ 60 MHz, you must configure $t_{HS\text{-}TRAIL}$ duration closer to but not exceeding t_{EOT} limit as defined by the MIPI D-PHY specification. This configuration ensures reliable data capture and proper end-of-transmission detection under low-frequency conditions.
- Some IP configurations may have slower Fmax when used in devices with slow performance grade.

Appendix A. Resource Utilization

The tables below show a sample resource utilization of the MIPI CSI/DSI IP on the LFCPNX-100-9LFG672C device using the Synplify Pro synthesis tool.

Table A.1. MIPI CSI/DSI Transmitter Resource Utilization[1]

IP Configuration	Slice	LUTs	Registers	EBR	High-Speed I/O Resources
PHY Direction == TX *PHY Mode == Soft D-PHY* *PHY Data Width* == 8 *Number of D-PHY Lanes* == 4 *Line Rate Per Lane* == 1000 *PHY Clock Mode == Continuous* *Protocol Mode == CSI-2* *Line Buffer Depth == 512* *Number of Pixel per Clock (PPC)* == 4 *Enable RGB888 Color Format* == checked	2335/79872	1769/79872	2335/80769	8/208	5 x ODDRX4, 1 x ECLKDIV, 1 x ECLKSYNC
PHY Direction == TX *PHY Mode == Soft D-PHY* *PHY Data Width* == 8 *Number of D-PHY Lanes* == 2 *Line Rate Per Lane* == 1500 *PHY Clock Mode == Continuous* *Protocol Mode == CSI-2* *Line Buffer Depth == 512* *Number of Pixel per Clock (PPC)* == 4 *Enable RGB888 Color Format* == checked	2288/79872	1747/79872	2288/80769	8/208	3 x ODDRX4, 1 x ECLKDIV, 1 x ECLKSYNC
PHY Direction == TX *PHY Mode == Soft D-PHY* *PHY Data Width* == 8 *Number of D-PHY Lanes* == 4 *Line Rate Per Lane* == 1000 *PHY Clock Mode == Continuous* *Protocol Mode == CSI-2* *Line Buffer Depth == 512* *Number of Pixel per Clock (PPC)* == 4 *Enable RGB565 Color Format* == checked	1677/79872	1071/79872	1677/80769	8/208	5 x ODDRX4, 1 x ECLKDIV, 1 x ECLKSYNC
PHY Direction == TX *PHY Mode == Soft D-PHY* *PHY Data Width* == 8 *Number of D-PHY Lanes* == 4 *Line Rate Per Lane* == 1500 *PHY Clock Mode == Non-Continuous* *Protocol Mode == CSI-2* *Line Buffer Depth == 2048* *Number of Pixel per Clock (PPC)* == 1 *Enable RGB888 Color Format* == checked	2361/79872	1751/79872	2361/80769	32/208	5 x ODDRX4, 1 x ECLKDIV, 1 x ECLKSYNC

IP Configuration	Slice	LUTs	Registers	EBR	High-Speed I/O Resources
PHY Direction == *TX* *PHY Mode* == *Soft D-PHY* *PHY Data Width* == 8 *Number of D-PHY Lanes* == 4 *Line Rate Per Lane* == 1000 *PHY Clock Mode* == *Continuous* *Protocol Mode* == *CSI-2* *Line Buffer Depth* == *512* *Number of Pixel per Clock (PPC)* == 4 *Enable RAW10 Color Format* == checked	2760/79872	2760/79872	2760/80769	8/208	5 x ODDRX4, 1 x ECLKDIV, 1 x ECLKSYNC

Note:

1. All other settings are default.

Table A.2. MIPI CSI/DSI Receiver Resource Utilization[1]

IP Configuration	Slices	LUTs	Registers	EBR	High-Speed I/O Resources
PHY Direction == *RX* *PHY Mode* == *Soft D-PHY* *PHY Data Width* == 8 *Number of D-PHY Lanes* == 4 *Line Rate Per Lane* == 1000 *PHY Clock Mode* == *Continuous* *Protocol Mode* == *DSI-2* *Number of Pixel per Clock (PPC)* == 4 *Enable RGB888 Color Format* == checked	1646/79872	1828/79872	1647/80769	8/208	4 x IDDRX4, 1 x ECLKDIV, 1 x ECLKSYNC
PHY Direction == *RX* *PHY Mode* == *Soft D-PHY* *PHY Data Width* == 8 *Number of D-PHY Lanes* == 2 *Line Rate Per Lane* == 1000 *PHY Clock Mode* == *Continuous* *Protocol Mode* == *DSI-2* *Number of Pixel per Clock (PPC)* == 4 *Enable RGB888 Color Format* == checked	1431/79872	1382/79872	1432/80769	6/208	2 x IDDRX4, 1 x ECLKDIV, 1 x ECLKSYNC
PHY Direction == *RX* *PHY Mode* == *Soft D-PHY* *PHY Data Width* == 8 *Number of D-PHY Lanes* == 4 *Line Rate Per Lane* == 1000 *PHY Clock Mode* == *Continuous* *Protocol Mode* == *CSI-2* *Number of Pixel per Clock (PPC)* == 4 *Enable RGB888 Color Format* == checked	1678/79872	1683/79872	1677/80769	8/208	4 x IDDRX4, 1 x ECLKDIV, 1 x ECLKSYNC
PHY Direction == *RX* *PHY Mode* == *Soft D-PHY* *PHY Data Width* == 8 *Number of D-PHY Lanes* == 4 *Line Rate Per Lane* == 1000 *PHY Clock Mode* == *Continuous* *Protocol Mode* == *CSI-2* *Number of Pixel per Clock (PPC)* == 1 *Enable RGB888 Color Format* == checked	1422/79872	1694/79872	1423/80769	6/208	4 x IDDRX4, 1 x ECLKDIV, 1 x ECLKSYNC

IP Configuration	Slices	LUTs	Registers	EBR	High-Speed I/O Resources
PHY Direction == RX *PHY Mode == Soft D-PHY* *PHY Data Width* == 8 *Number of D-PHY Lanes* == 4 *Line Rate Per Lane* == 1000 *PHY Clock Mode == Continuous* *Protocol Mode == CSI-2* *Number of Pixel per Clock (PPC)* == 4 *Enable RAW10 Color Format* == checked	1705/79872	1786/79872	1706/80769	8/208	4 x IDDRX4, 1 x ECLKDIV, 1 x ECLKSYNC

Note:

1. All other settings are default.

For more information about specific configuration, generate the IP, run synthesis and MAP, and check the MAP reports for resource utilization. Numbers may vary when using a different software version, device density, synthesis tool, or performance grade.

References

- MIPI CSI/DSI IP Release Notes (FPGA-RN-02112)
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- Certus-NX web page
- CertusPro-NX web page
- CrossLink-NX web page
- MachXO5-NX web page
- Lattice Radiant Software web page
- Lattice Propel Design Environment web page
- Lattice Solutions IP Cores web page
- Lattice Solutions Reference Designs web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

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

Revision History

Note: In some instances, the IP may be updated without changes to the user guide. The user guide may reflect an earlier IP version but remains fully compatible with the later IP version. Refer to the IP Release Notes for the latest updates.

Revision 1.0, IP v4.0.0, April 2026

Section	Change Summary
All	Initial release.

www.latticesemi.com