# SLVS-EC Receiver IP

IP Version: v1.2.4

# User Guide

FPGA-IPUG-02125-2.1

October 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.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# Contents

Contents ..... 3
Abbreviations in This Document ..... 7
1. Introduction ..... 8
1.1. Overview of the IP ..... 8
1.2. Quick Facts ..... 8
1.3. IP Support Summary ..... 8
1.4. Features ..... 8
1.5. Licensing and Ordering Information ..... 8
1.5.1. Ordering Part Number ..... 9
1.6. Minimum Device Requirements ..... 9
1.7. Naming Conventions ..... 9
1.7.1. Nomenclature ..... 9
1.7.2. Signal Names ..... 9
1.7.3. Attribute Names ..... 9
2. Functional Description ..... 10
2.1. IP Architecture Overview ..... 10
2.2. Clocking ..... 11
2.2.1. Clocking Overview ..... 11
2.2.2. Adjusting Reference Clock for Different Baud Rate ..... 11
2.3. Reset ..... 12
2.4. User Interfaces ..... 13
2.5. MPCS Soft IP Wrapper ..... 14
2.6. SLVS-EC Rx Core ..... 14
2.6.1. Rx Protocol Management FSM ..... 14
2.6.2. Packet Parser ..... 16
2.6.3. Byte-to-Pixel Converter ..... 16
2.7. Configuration, Control, and Status Registers (CSR) ..... 19
2.8. Timing Descriptions ..... 19
2.8.1. SLVS-EC Rx with Native Interface ..... 19
2.8.2. SLVS-EC Rx with AXI4-Stream Transmitter Interface ..... 20
2.8.3. SLVS-EC Receiver UVSI ..... 21
3. IP Parameter Description ..... 25
3.1. General ..... 25
4. Signal Description ..... 27
4.1. Clock and Reset ..... 27
4.2. Primary I/O ..... 27
4.3. Native Interface ..... 27
4.4. Packet Header Information ..... 28
4.5. Control and Status ..... 28
4.6. Miscellaneous ..... 29
4.7. AXI4-Stream Transmitter Interface ..... 29
4.8. Unified Video Streaming Interface ..... 29
4.9. APB Interface ..... 30
5. Register Description ..... 31
6. Designing with the IP ..... 34
6.1. Generating and Instantiating the IP ..... 34
6.1.1. Generated Files and File Structure ..... 36
6.2. Design Implementation ..... 37
6.3. Timing and Physical Design Constraints ..... 37
6.4. Specifying the Strategy ..... 38
6.5. Running Functional Simulation ..... 38
7. Debugging ..... 42

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

7.1. Debug Tools ..... 42
7.1.1. Reveal Analyzer ..... 42
7.1.2. QuestaSim ..... 42
Appendix A. Resource Utilization ..... 43
Appendix B. Limitations ..... 44
References ..... 45
Technical Support Assistance ..... 46
Revision History ..... 47

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# Figures

Figure 2.1. Lattice SLVS-EC Receiver Block Diagram ..... 10
Figure 2.2. Clock Domain Crossing Diagram ..... 11
Figure 2.3. Control Sequence for Initialization ..... 13
Figure 2.4. Training Sequence Details ..... 13
Figure 2.5. Rx Protocol Management FSM ..... 14
Figure 2.6. Input/Output Behavior of Rx Protocol Management FSM Module ..... 15
Figure 2.7. Byte-to-Pixel Converter Implementation ..... 16
Figure 2.8. FIFO Implementation Diagram ..... 17
Figure 2.9. Configuration, Control, and Status Register Implementation ..... 19
Figure 2.10. SLVS-EC Rx Native Interface Output Timing Diagram ..... 20
Figure 2.11. SLVS-EC Rx AXI4-Stream Interface Timing Diagram ..... 21
Figure 2.12. UVSI Rx Timing Diagram without TREADY Deassertion ..... 21
Figure 2.13. UVSI Rx Timing Diagram with TREADY Deassertion ..... 22
Figure 2.14. TDATA Pixel Mapping for Sample RAW8 Configuration ..... 22
Figure 2.15. TDATA Pixel Mapping for Sample RAW10 Configuration ..... 23
Figure 2.16. TDATA Pixel Mapping for Sample RAW12 Configuration ..... 23
Figure 2.17. TDATA Pixel Mapping for Sample RAW14 Configuration ..... 23
Figure 2.18. TDATA Pixel Mapping for Sample RAW16 Configuration ..... 24
Figure 6.1. Module/IP Block Wizard ..... 34
Figure 6.2. IP Configuration ..... 35
Figure 6.3. Check Generated Result ..... 36
Figure 6.4. Simulation Wizard ..... 38
Figure 6.5. Add and Reorder Source ..... 39
Figure 6.6. Parse HDL files for simulation ..... 39
Figure 6.7. Summary ..... 40
Figure 6.8. Simulation Waveform ..... 40
Figure 6.9. Simulation Log on a Complete Run ..... 41

# Tables

Table 1.1. Summary of the SLVS-EC Receiver IP ..... 8
Table 1.2. SLVS-EC Receiver IP Support Readiness ..... 8
Table 1.3. Ordering Part Number ..... 9
Table 2.1. Pre-Configured MPCS PLL Setting for SLVS-EC Rx ..... 11
Table 2.2. User Interfaces and Supported Protocols ..... 13
Table 2.3. Byte-to-Pixel Converter Data Groups ..... 16
Table 2.4. TDATA Mapping per Data Type ..... 22
Table 3.1. General Attributes ..... 25
Table 4.1. Clock and Reset Ports ..... 27
Table 4.2. Primary I/O Ports ..... 27
Table 4.3. Native Interface Ports ..... 27
Table 4.4. Byte-to-Pixel Data Width by Baud Grade and Number of Rx Lanes ..... 28
Table 4.5. Packet Header Information Ports ..... 28
Table 4.6. Control and Status Ports ..... 28
Table 4.7. Miscellaneous Ports ..... 29
Table 4.8. AXI4-Stream Transmitter Interface Ports ..... 29
Table 4.9. Unified Video Streaming Interface Ports ..... 29
Table 4.10. APB Interface Ports ..... 30
Table 5.1. Access Type Definition ..... 31

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Table 5.2. Summary of SLVS-EC Receiver Registers ..... 31
Table 6.1. Generated File List ..... 36
Table A.1. Resource Utilization ..... 43

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# Abbreviations in This Document

A list of abbreviations used in this document.

| Abbreviation | Definition |
|---|---|
| AMBA | Advanced Microcontroller Bus Architecture |
| APB | Advanced Peripheral Bus |
| AXI | Advanced eXtensible Interface |
| CIS | CMOS Image Sensor |
| CMOS | Complementary Metal-Oxide Semiconductor |
| CRC | Cyclic Redundancy Check |
| CSR | Configuration, Control and Status Register |
| DMA | Direct Memory Access |
| DSP | Digital Signal Processor |
| DUT | Device Under Test |
| EBR | Embedded Block RAM |
| ECC | Error Correction Code |
| FIFO | First-In First-Out |
| FPGA | Field-Programmable Gate Array |
| FSM | Finite State Machine |
| GUI | Graphical User Interface |
| HDL | Hardware Description Language |
| I/O | Input/Output |
| IP | Intellectual Property |
| JTAG | Joint Test Action Group |
| LMMI | Lattice Memory Mapped Interface |
| LSB | Least Significant Bit |
| LUT | Look-Up Table |
| MPCS | Multi-Protocol PCS |
| OEM | Original Equipment Manufacturer |
| PCS | Physical Coding Sublayer |
| PDC | Post-synthesis Design Constraint |
| PHY | Physical |
| PLL | Phase-Locked Loop |
| PMA | Physical Media Attachment |
| RO | Read-Only |
| RSVD | Reserved |
| RTL | Register Transfer Level |
| RW | Read-Write |
| RW1C | Read-Write 1 to Clear |
| Rx | Receiver |
| SERDES | Serializer Deserializer |
| SLVS-EC | Scalable Low Voltage Signaling Embedded Clock |
| SOF | Start of Frame |
| Tx | Transmitter |
| UVSI | Unified Video Streaming Interface |
| WO | Write-Only |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# 1. Introduction

## 1.1. Overview of the IP

This user guide describes the Lattice Scalable Low Voltage Signaling-Embedded Clock Receiver (SLVS-EC Rx) IP. The SLVS-EC Receiver IP provides the FPGA an interface to receive serial data from CMOS Image Sensors and offers a solution to convert the incoming serial data to a parallel pixel data format. The IP can receive up to eight lanes of differential serial data running at a maximum of 5 Gbps per lane. The SLVS-EC functions defined by the PHY Layer are supported by the Multi-Protocol PCS (MPCS) soft IP which instantiates the PMA/PCS IPs. The SLVS-EC IP also supports the Link Layer functions defined by the PHY Layer.

## 1.2. Quick Facts

**Table 1.1. Summary of the SLVS-EC Receiver IP**

| | | |
|---|---|---|
| **IP Requirements** | Supported Devices | CertusPro™-NX |
| | IP Changes[1] | Refer to the SLVS-EC Receiver Release Notes (FPGA-RN-02121). |
| **Resource Utilization** | Supported User Interface | AXI4-Stream Interface, Unified Video Streaming Interface |
| | Resources | Refer to Appendix A. Resource Utilization. |
| **Design Tool Support** | Lattice Implementation | IP Core v1.2.4 – Lattice Radiant™ software 2026.1.1 |
| | Synthesis | Lattice Synthesis Engine |
| | | Synopsys Synplify Pro® for Lattice |
| | Simulation | For a list of supported simulators, see the Lattice Radiant Software user guide. |

**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. SLVS-EC Receiver IP Support Readiness**

| Device Family | PHY Direction | Baud Grade | Line Rate per Lane (Mbps) | Radiant Timing Model |
|---|---|---|---|---|
| CertusPro-NX | Rx | 1 | 742.5–1250 | Final |
| | | 2 | 1485–2500 | Final |
| | | 3 | 2970–5000 | Final |

## 1.4. Features

Key features of the SLVS-EC Receiver IP include:

- Compliant with SLVS-EC Protocol specification v2.0.
- Backward compatibility with SLVS-EC Protocol specification v1.2.
- Supports PHY layer functions using the embedded PMA/PCS in CertusPro-NX FPGA device.
- Supports up to eight lanes each running at a maximum of 5 Gbps baud rate.
- Supports AXI4-Stream Protocol as a selectable option for data interface.
- Supports UVSI as a selectable option for data interface.
- Supports AMBA 3 APB Protocol v1.0 for register access to control the IP.

## 1.5. Licensing and Ordering Information

An IP-specific license string is required to enable full use of the SLVS-EC Receiver IP in a complete, top-level design.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

The IP can be fully evaluated through functional simulation and implementation (synthesis, map, place, and route) without an IP license string. This IP supports Lattice's IP hardware evaluation capabilities. You can create versions of the IP to operate in hardware for a limited time (approximately four hours) without requiring an IP license string. A license string is required to enable timing simulation and to generate a bitstream file that does not include the hardware evaluation timeout limitation.

For more information about pricing and availability of the SLVS-EC Receiver IP, contact your local Lattice Sales Office.

### 1.5.1. Ordering Part Number

**Table 1.3. Ordering Part Number**

| Device Family | Part Number | |
|---|---|---|
| | Single Seat Annual | Single Seat Perpetual |
| CertusPro-NX | SLVS-EC-RX-CPNX-US | SLVS-EC-RX-CPNX-UT |

## 1.6. Minimum Device Requirements

Refer to the Appendix A. Resource Utilization section for the minimum required resources 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

- _*n* are active low (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*).

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# 2. Functional Description

## 2.1. IP Architecture Overview

SLVS-EC is a high-speed serial interface technology developed by Sony for the next-generation high-resolution CMOS Image Sensors. The interface provides a unidirectional wideband pixel data transfer from a CMOS Image Sensor (CIS) to a Digital Signal Processor (DSP).

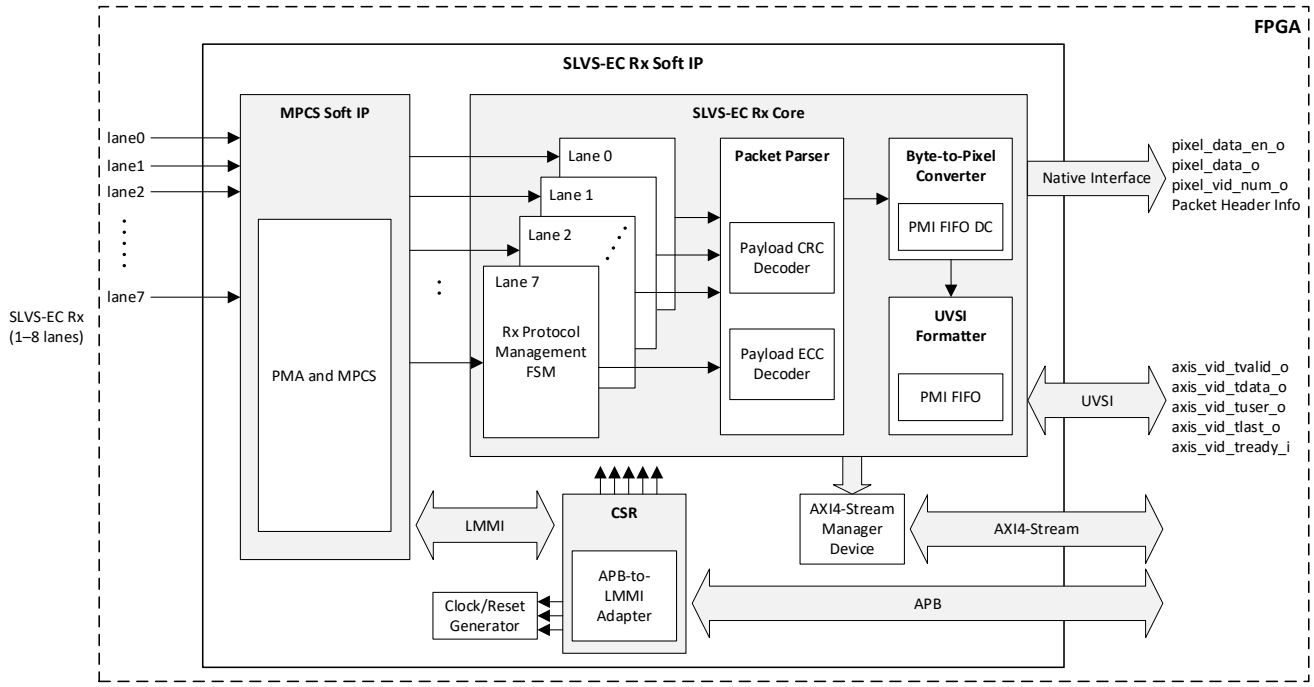



**Figure 2.1. Lattice SLVS-EC Receiver Block Diagram**

The SLVS-EC Receiver IP includes the following submodules:

- MPCS Soft IP Wrapper
- SLVS-EC Rx Core
- Transaction Layer

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

## 2.2. Clocking

### 2.2.1. Clocking Overview

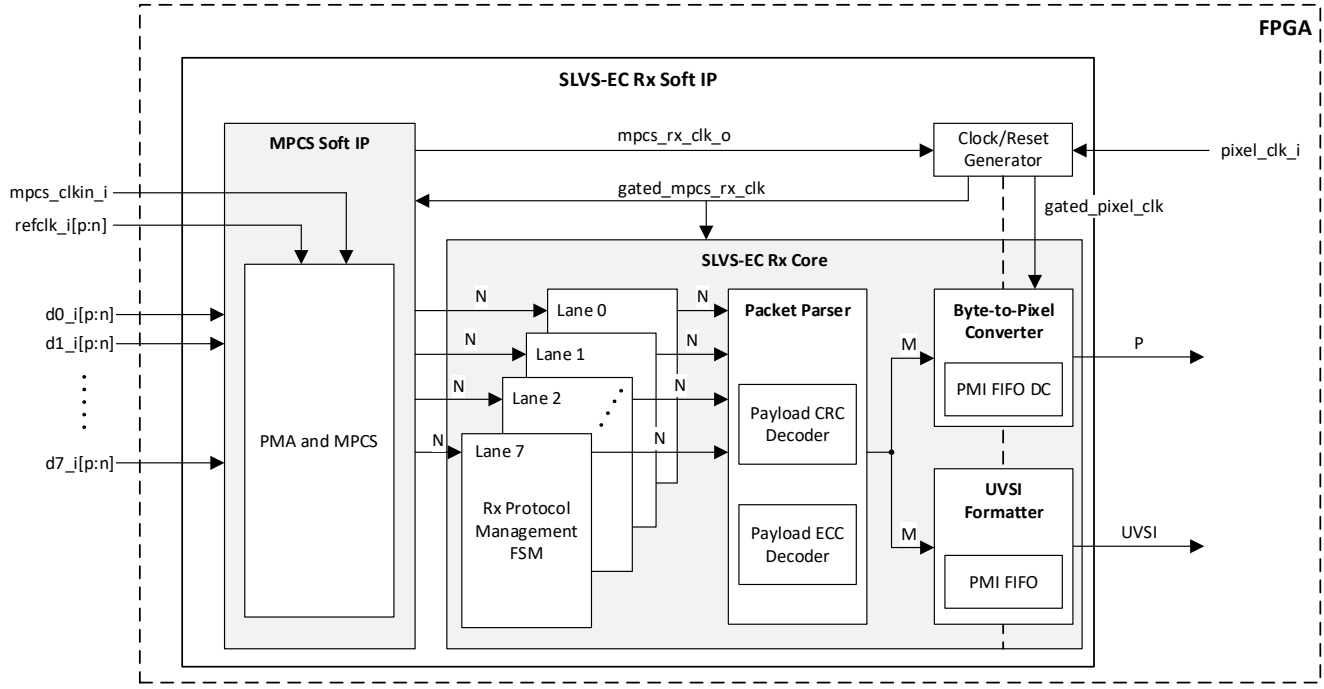



**Figure 2.2. Clock Domain Crossing Diagram**

Figure 2.2 shows the clocking scheme for the SLVS-EC Receiver soft IP. The output clock (mpcs_rx_clk_o) generated by the MPCS soft IP clocks the Packet Parser module and the byte domain side of Byte-to-Pixel Converter and UVSI Formatter (only one of these blocks is active at a time, depending on whether the Native or UVSI interface is selected). Each of these two blocks contains its own internal FIFO to handle the clock domain crossing between mpcs_rx_clk_o and pixel_clk_i.

The MPCS Rx clock frequency depends on the reference clock used as shown in Table 2.1. The Pixel clock frequency varies depending on the configuration (that is, the number of Output Pixels, Data Type, and FIFO setting).

### 2.2.2. Adjusting Reference Clock for Different Baud Rate

The MPCS PLL settings for each Baud Grade are pre-configured for SLVS-EC as shown in Table 2.1.

**Table 2.1. Pre-Configured MPCS PLL Setting for SLVS-EC Rx**

| Baud Grade | Bit Rate (Mbps) | Reference Clock (MHz)[1] | M | F | N | $F_{VCO}$ (MHz) | $F_{bit}$ (MHz) | $F_{PMA}$ (MHz) | MPCS Rx Clock (MHz) |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 5000 | 125 | 2 | 4 | 10 | 10000 | 5000 | 500 | 125 |
| 2 | 2500 | 125 | 4 | 2 | 10 | 10000 | 2500 | 250 | 125 |
| 1 | 1200 | 125 | 8 | 1 | 10 | 10000 | 1250 | 125 | 125 |

**Note:**

1. Frequency of Reference Clock can be varied depending on the target Bit Rate for each Baud Grade.

Simplified equations based on PLL settings are shown below:

$$Internal\ Data\ Width\ Factor\ (bits) = 10\ (Baud\ Grade\ 1);\ 20\ (Baud\ Grade\ 2);\ 40\ (Baud\ Grade\ 3)$$

$$Reference\ Clock\ (MHz) = \frac{Target\ Baud\ Rate}{Internal\ Data\ Width\ Factor}$$

$$MPCS\ Rx\ Clock\ (MHz) = Reference\ Clock$$

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Example:

Case 1: Baud Grade 1, Target baud rate of 1250 Mbps

$$Internal\ Data\ Width\ Factor\ (bits) = 10$$
$$Reference\ Clock\ (MHz) = \frac{1250\ Mbps}{10\ bits} = 125\ MHz$$
$$MPCS\ Rx\ Clock\ (MHz) = 125\ MHz$$

In the scenario when a different baud rate is desired, the frequency of reference clock (refclk_n_i/refclk_p_i) can be adjusted. To do so, the target Baud Grade must be selected first in the IP user interface. To compute for the frequency of reference clock to be used, follow the equations above. Note, however, that only a range of reference clock frequencies is supported by MPCS/SERDES. For details, refer to CertusPro-NX SERDES/PCS User Guide (FPGA-TN-02245).

Example:

Case 2: Baud Grade 2, Target baud rate of 2304 Mbps

$$Internal\ Data\ Width\ Factor\ (bits) = 20$$
$$Reference\ Clock\ (MHz) = \frac{2304\ Mbps}{20\ bits} = 115.2\ MHz$$
$$MPCS\ Rx\ Clock\ (MHz) = 115.2\ MHz$$

Case 3: Baud Grade 3, Target baud rate of 4752 Mbps

$$Internal\ Data\ Width\ Factor\ (bits) = 40$$
$$Reference\ Clock\ (MHz) = \frac{4752\ Mbps}{40\ bits} = 118.8\ MHz$$
$$MPCS\ Rx\ Clock\ (MHz) = 118.8\ MHz$$

## 2.3. Reset

Listed below are the reset and initialization flows for the SLVS-EC Receiver IP:

- Reset
  An asynchronous reset pin (active low) as system reset is used for resetting the SLVS-EC Receiver Soft IP. Internal reset logic is implemented to guarantee synchronous deassertion throughout the different clock domains among the MPCS soft IP and other soft logic blocks.
- Initialization
  Figure 2.3 shows the control sequence needed for the PHY initialization of SLVS-EC Receiver. Toggle mpcs_clkin_i, serial reference clock and wait for mpcs_ready_o to assert, to start the training sequence. A setup signal is sent from the client side to the SLVS-EC Rx for it to commence start-up and to be ready to receive the initialization sequence. CIS drives the link low before sending the training sequence. This training sequence is a series of repeated Sync Codes and intermittent transmission of Deskew codes, to establish bit synchronization and adjust the skew between the lanes respectively. See Figure 2.4 for a detailed definition of the training sequence. The PHY control codes, number of transfer times, and other training sequence items are to be set beforehand through APB register configuration (see Table 5.2 for the list of attribute registers). After the training sequence has been received and processed successfully by the SLVS-EC Receiver, the SLVS-EC Receiver sends a ready signal to the client side to confirm that it is already in the state of receiving valid packets of data.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

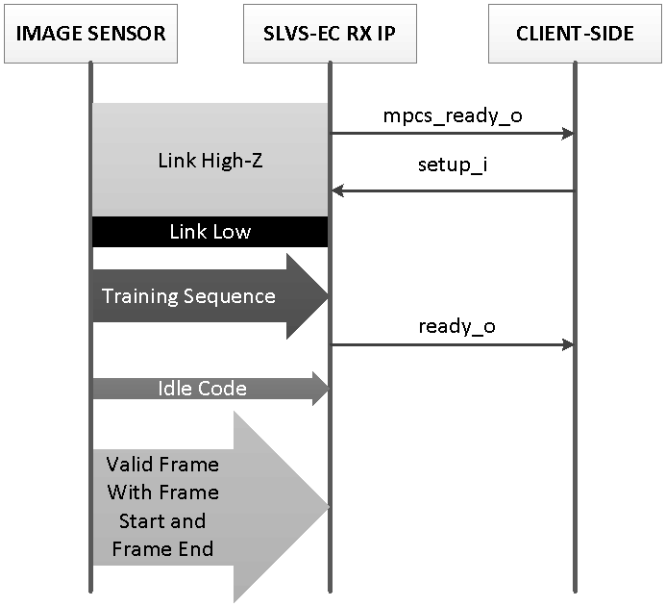



**Figure 2.3. Control Sequence for Initialization**

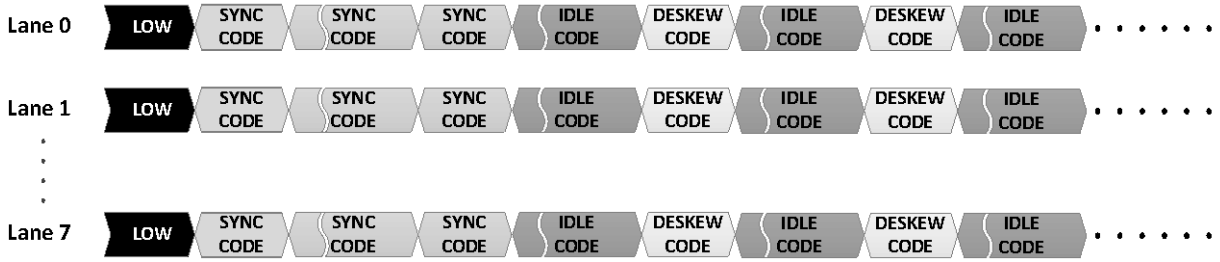



**Figure 2.4. Training Sequence Details**

## 2.4. User Interfaces

**Table 2.2. User Interfaces and Supported Protocols**

| User Interface | Supported Protocols | Description |
|---|---|---|
| Device Transmitter | AXI4-Stream | The AXI4-Stream interface carries a stream of pixel data and packet header. The packet header information is considered as user sideband signals (m_tuser_o) if header analysis is enabled. |
| Device Transmitter | UVSI | The UVSI transfers video data in the pixel domain and complies with the AXI4-Stream protocol. 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, provided TREADY deassertion does not exceed the fixed 128-word buffer depth. For the complete list of interface signals, refer to the Signal |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

| User Interface | Supported Protocols | Description |
|---|---|---|
| | | Description section. |
| Control | APB | Used to configure IP CSRs, including MPCS LMMI registers. Refer to the Register Description section for more information. |

## 2.5. MPCS Soft IP Wrapper

MPCS Soft IP supports a wide range of serial I/O protocols including the SLVS-EC. The PMA and PCS settings in this IP are pre-configured for SLVS-EC. This IP receives the serial data and performs mainly the word alignment, 8b/10b decoding, and lane-to-lane Rx deskew of the incoming data. Refer to the MPCS Module User Guide (FPGA-IPUG-02118) for further details.

## 2.6. SLVS-EC Rx Core

The SLVS-EC Rx Core, consisting of Rx Protocol Management FSM, Packet Parser, and Byte-to-Pixel Converter modules, receives the decoded byte data from the MPCS soft IP and handles the IP Link Layer functions such as packet header and packet footer processing, link protocol management, and data formatting.

### 2.6.1. Rx Protocol Management FSM

This module receives the parallel data from the MPCS soft IP wrapper and strips off the PHY control codes. Figure 2.5 is the finite state machine diagram of the Rx Protocol Management FSM module. The FSM on the left side of the diagram is for the training sequence and on the right side of the diagram is the main FSM to determine the PHY control codes. The output of this module contains the packet header, valid payload, and payload/lane stuffing bytes, if there are any, as shown in Figure 2.6.

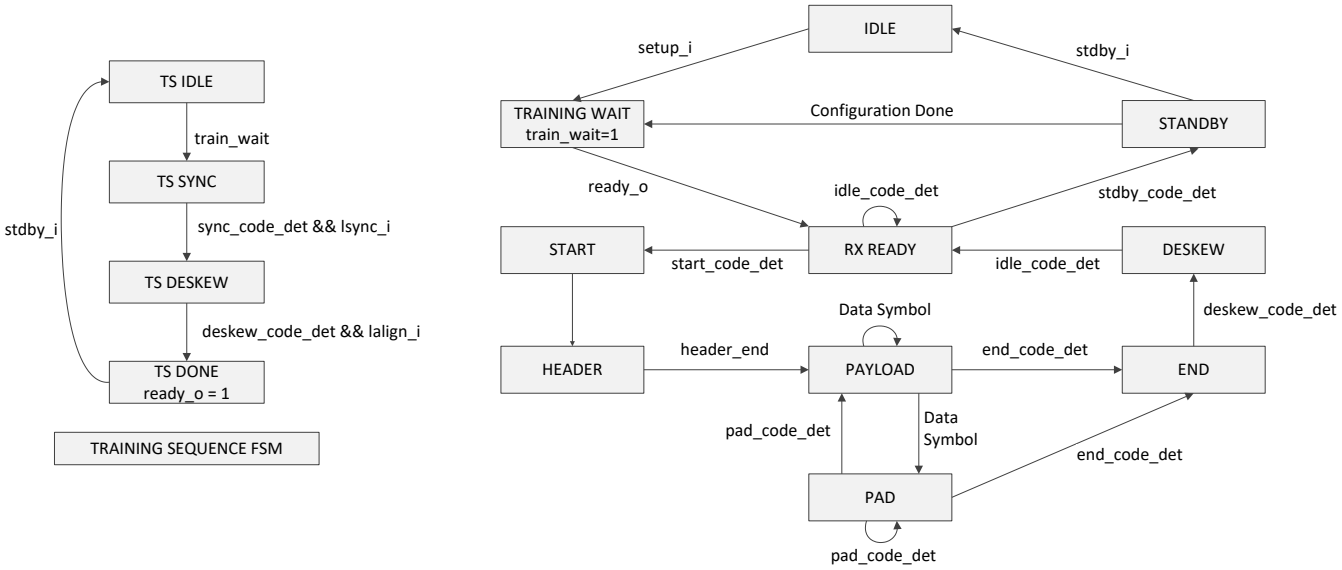



**Figure 2.5. Rx Protocol Management FSM**

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

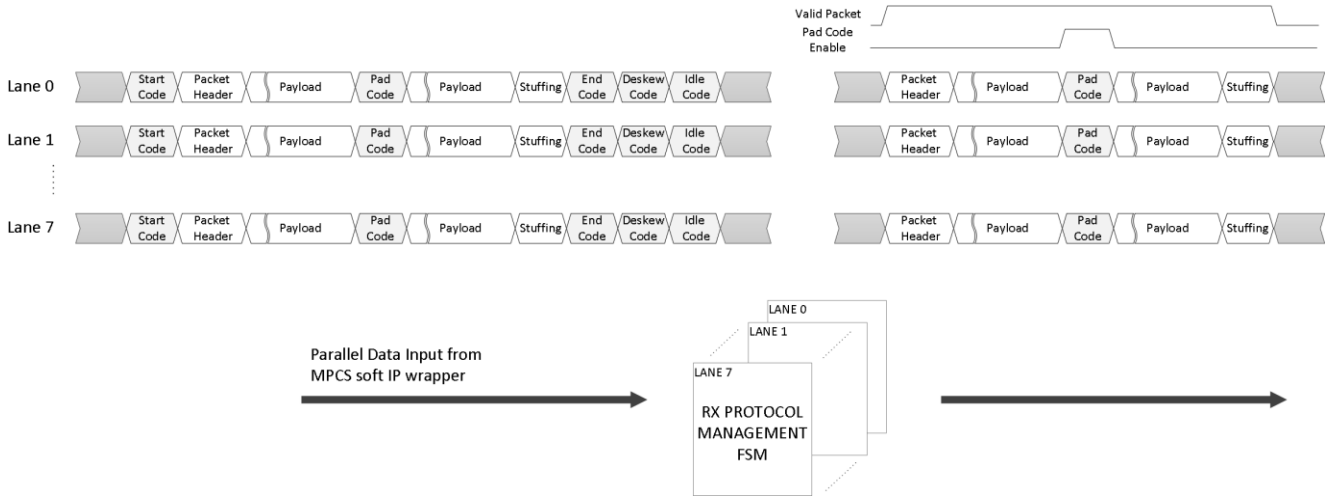



**Figure 2.6. Input/Output Behavior of Rx Protocol Management FSM Module**

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

The current IP supports lane deskew only during the initialization/training phase. In cases when lane skew occurs during packet data transfer, you can perform the following procedure:

1. In the SLVS-EC Rx soft IP, add custom deskew logic after the MPCS IP instance that searches for the START code from packet data of each lane.
2. Add a single-clock FIFO that stores these packets after START is detected.
3. Read out from FIFO after START is detected in all lanes.

### 2.6.2. Packet Parser

This module extracts the parallel payload data and parses the packet header and packet footer information of the received packet from the Rx Protocol Management FSM module. The Payload CRC decoder and Payload ECC decoder are also instantiated in this module for any payload error detection.

### 2.6.3. Byte-to-Pixel Converter

This module converts the received valid line data bytes from the Packet Parser module into pixel data format. The PMI FIFO DC is used to transfer the valid line data from all lanes from the MPCS Rx clock domain (mpcs_rx_clk_o) to the pixel clock domain (pixel_clk_i). Figure 2.7 shows the diagram of the Byte-to-Pixel Converter Implementation. The input data is pipelined and grouped depending on the Data Type configuration to extract pixel data in the pixel domain (see Table 2.3).

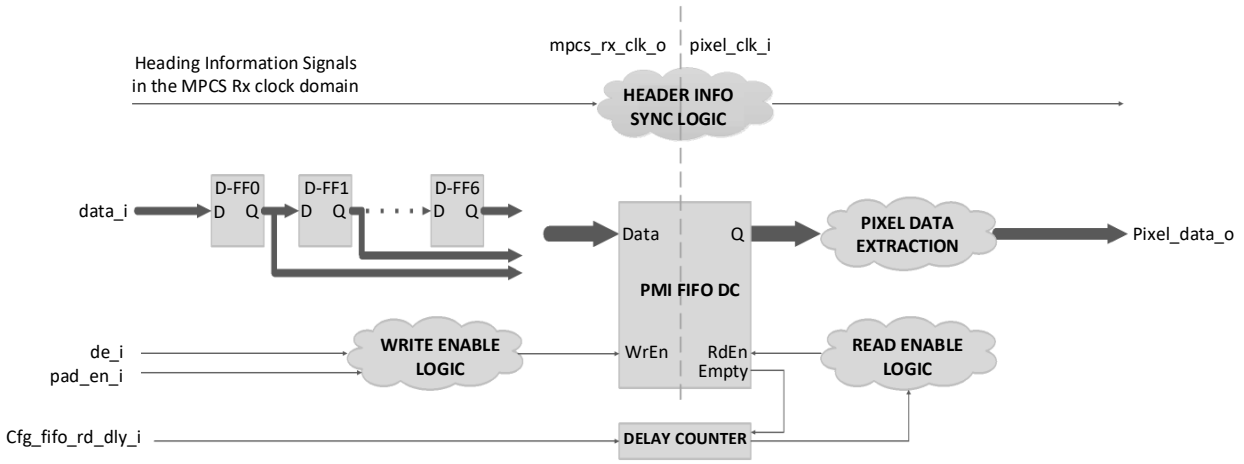



**Figure 2.7. Byte-to-Pixel Converter Implementation**

**Table 2.3. Byte-to-Pixel Converter Data Groups**

| Data Type | Data Groups | Pixel Group |
|---|---|---|
| RAW 8 | 1 | 1 |
| RAW 10 | 5 | 4 |
| RAW 12 | 3 | 2 |
| RAW 14 | 7 | 4 |
| RAW 16 | 2 | 1: For Baud 1 and Lane 1<br>2: Else |

The FIFO depth is defined based on the design configuration, as shown in the Attributes List in Table 3.1. To avoid the FIFO full or empty states, the Data Safe Zone is defined by setting the data FIFO Delay attribute, which handles the Overflow/Underflow Threshold, as shown in Figure 2.8. The FIFO delay value is used to trigger the start of FIFO read-out from FIFO not-empty state.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

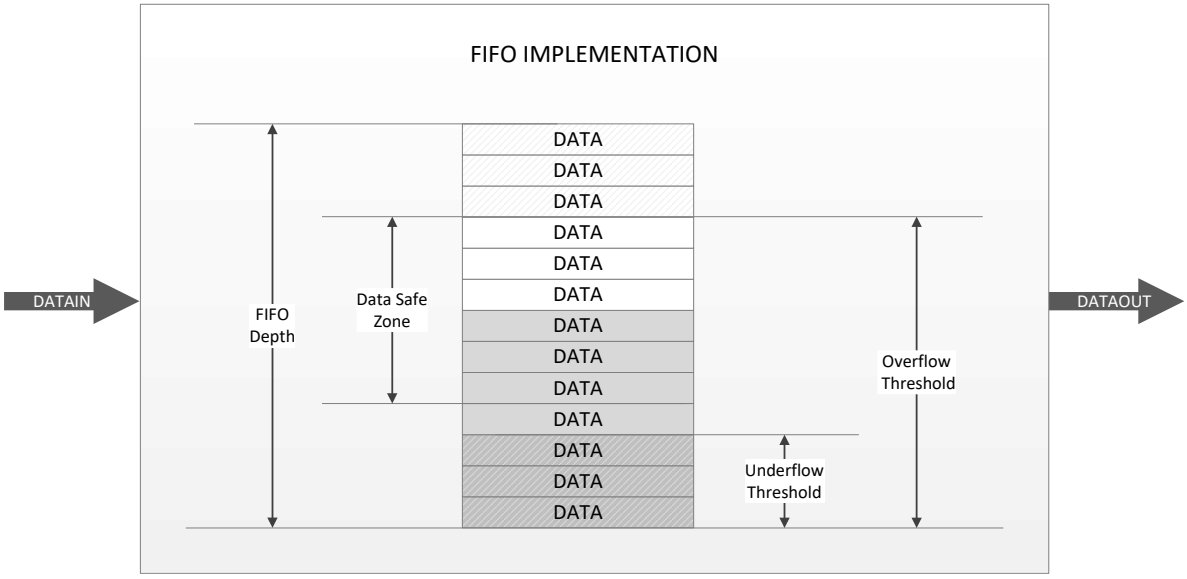



**Figure 2.8. FIFO Implementation Diagram**

The FIFO has the following three attributes: FIFO Width, FIFO Depth, and FIFO Delay. The FIFO Delay is used to synchronize data flows between byte and pixel sides. Data transfer on the pixel side starts after the FIFO Delay is reached. FIFO Width, FIFO Depth, and FIFO Delay depend on several attributes listed below:

- Number of Rx Lanes (see Table 3.1)
- MPCS Output Data Width (based on the Baud Grade setting)
- Baud Grade 1: 8 bits
- Baud Grade 2: 16 bits
- Baud Grade 3: 32 bits
- MPCS Rx Clock Frequency (mpcs_rx_clk_o, frequency is the same as the frequency of the reference clock ref_clk_p/n_i used)
- Pixel Clock Frequency (pixel_clk_i, available on the user interface)
- Byte Count (converted from Pixel Count – see Table 3.1)
- Number of Bits Per Pixel (Data Type – see Table 3.1)
- Number of Output Pixels per Clock (see Table 3.1)

FIFO Width is calculated as,

$$FIFO\ Width = 7 \times Number\ of\ Rx\ Lanes \times MPCS\ Output\ Data\ Width$$

The input data must be grouped based on the pixel to byte conversion multiplier in order to properly extract the pixel data in the pixel clock domain.

Denote the Data Flow Speed on the byte side as MPCS Eff Rate and the Data Flow Speed on the pixel side as Pixel Eff Rate. They are calculated as,

$$MPCS\ Eff\ Rate = MPCS\ Rx\ Clock\ Frequency \times Number\ of\ Rx\ Lanes \times MPCS\ Output\ Data\ Width$$

$$Pixel\ Eff\ Rate = Pixel\ Clock\ Frequency \times Number\ of\ Bits\ Per\ Pixel \times Number\ of\ Output\ Pixels\ per\ Clock$$

The FIFO Depth (minimum value) and FIFO Delay depend on the ratio ($N_{ratio}$) of the Data Flow Speed on the Byte side and Data Flow Speed on the Pixel side. Pixel Eff Rate should be greater than or equal to MPCS Eff Rate.

$$N_{ratio} = Pixel\ Eff\ Rate / MPCS\ Eff\ Rate$$

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

They can be calculated by the following formulas:
IF (MPCS Eff Rate == Pixel Eff Rate)

$$FIFO\ Depth = 16; FIFO\ Delay = 4$$

ELSE

$$FIFO\ Delay\ Temp = Ceiling[\frac{(Nratio - 1)(Byte\ Count \times 8)}{No.of\ Lanes\ \times\ MPCS\ Output\ Data\ Width} \times \frac{1}{Data\ Group}]$$

$$FIFO\ Delay\ = (FIFO\ Delay\ Temp + 1)\ \times\ Pixel\ Group$$

$$FIFO\ Depth = 2^{clog_2(Fifo\ Delay\ Temp+4)}$$

If there are pad codes, add them to the Byte Count and recompute the new FIFO Delay/Depth (for example, byte count is 12,288 and additional 1,200 bytes of pad codes per line, new byte count = 12,288 + 1,200 = 13,488).

Sample Computations:

Case 1: Baud Grade = 2, Number of Rx Lanes = 8, Data Type = RAW10, MPCS Rx Clock = 115.2 MHz, Pixel Clock = 125 MHz, Pixel Count = 8432

MPCS Settings:

$$MPCS\ Output\ Data\ Width = 16\ bits\ (Baud\ Grade\ 2)$$
$$MPCS\ Output\ Data\ (bytes) = 2$$

Pixel Settings:

$$Output\ Pixels\ per\ clock = 16\ (\text{Table 3.1})$$
$$Pixel\ count\ per\ line\ (bytes) = 10540\ (Byte\ Count)$$
$$Data\ Group = 5\ (\text{Table 2.3})$$

Computations:

$$FIFO\ Width = 7 \times 8\ lanes \times 16\ bits = 896\ bits$$
$$MPCS\ Eff\ Rate = 115.2\ MHz \times 8\ lanes \times 16\ bits = 14{,}745.6\ Mbps$$
$$Pixel\ Eff\ Rate = 125\ MHz \times 10\ bits \times 16\ output\ pixels\ per\ clock = 20{,}000\ Mbps$$
$$N_{ratio} = \frac{20{,}000\ Mbps}{14{,}745.6\ Mbps} = 1.356336806$$

Since MPCS Eff Rate is not equal to Pixel Eff Rate:

$$FIFO\ Delay\ Temp = Ceiling\ [\frac{(1.356336806 - 1) \times (10540 \times 8)}{8 \times 16} \times \frac{1}{5}] = 47$$
$$FIFO\ Delay = (47 + 1) \times 5 = 240$$
$$FIFO\ Depth = 2^{clog_2\ (47+4)} = 64$$

Case 2: Baud Grade = 3, Number of Rx Lanes = 4, Data Type = RAW8, MPCS Rx Clock = 118.8 MHz, Pixel Clock = 125 MHz, Pixel Count = 1920

MPCS Settings:

$$MPCS\ Output\ Data\ Width = 32\ bits\ (Baud\ Grade\ 3)$$
$$MPCS\ Output\ Data\ (bytes) = 4$$

Pixel Settings:

$$Output\ Pixels\ per\ clock = 16\ (\text{Table 3.1})$$
$$Pixel\ count\ per\ line\ (bytes) = 1920\ (Byte\ Count)$$
$$Data\ Group = 1\ (\text{Table 2.3})$$

Computations:

$$FIFO\ Width = 7 \times 4\ lanes \times 32\ bits = 896\ bits$$
$$MPCS\ Eff\ Rate = 118.8\ MHz \times 4\ lanes\ \times 32\ bits = 15{,}206.4\ Mbps$$
$$Pixel\ Eff\ Rate = 125\ MHz \times 8\ bits \times 16\ output\ pixels\ per\ clock = 16{,}000\ Mbps$$
$$N_{ratio} = \frac{16{,}000\ Mbps}{15{,}206.4\ Mbps} = 1.052188552$$

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Since MPCS Eff Rate is not equal to Pixel Eff Rate:

$$FIFO\ Delay\ Temp = Ceiling\ [\frac{(1.052188552 - 1) \times (1920 \times 8)}{4 \times 32} \times \frac{1}{1}] = 7$$
$$FIFO\ Delay = (7 + 1) \times 1 = 8$$
$$FIFO\ Depth = 2^{clog_2\ (7+4)} = 16$$

## 2.7. Configuration, Control, and Status Registers (CSR)

Figure 2.9 shows the CSR Implementation that can be accessed through APB. MPCS registers can only be accessed through LMMI; thus, an APB-to-LMMI converter is needed and the SLVS-EC Rx IP registers are then mapped to the MPCS LMMI registers. It is required to start APB register access after at least 22 ns. The MPCS register initialization is done after this duration.

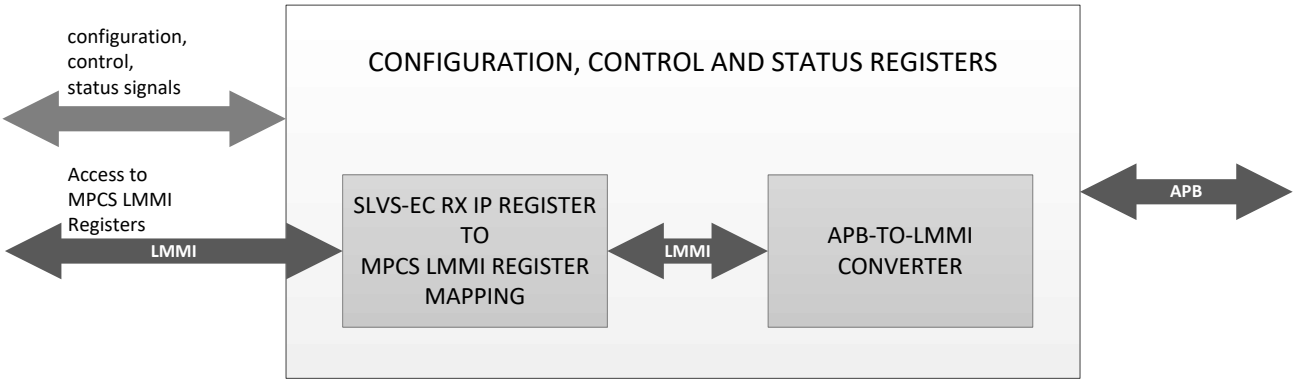



**Figure 2.9. Configuration, Control, and Status Register Implementation**

## 2.8. Timing Descriptions

### 2.8.1. SLVS-EC Rx with Native Interface

Figure 2.10 shows the output timing diagram of SLVS-EC Rx with Native Interface. The output signals carrying the packet header information are available at the output if the Header Analysis attribute is enabled. The packet footer information is available if the Miscellaneous Signals attribute is enabled.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

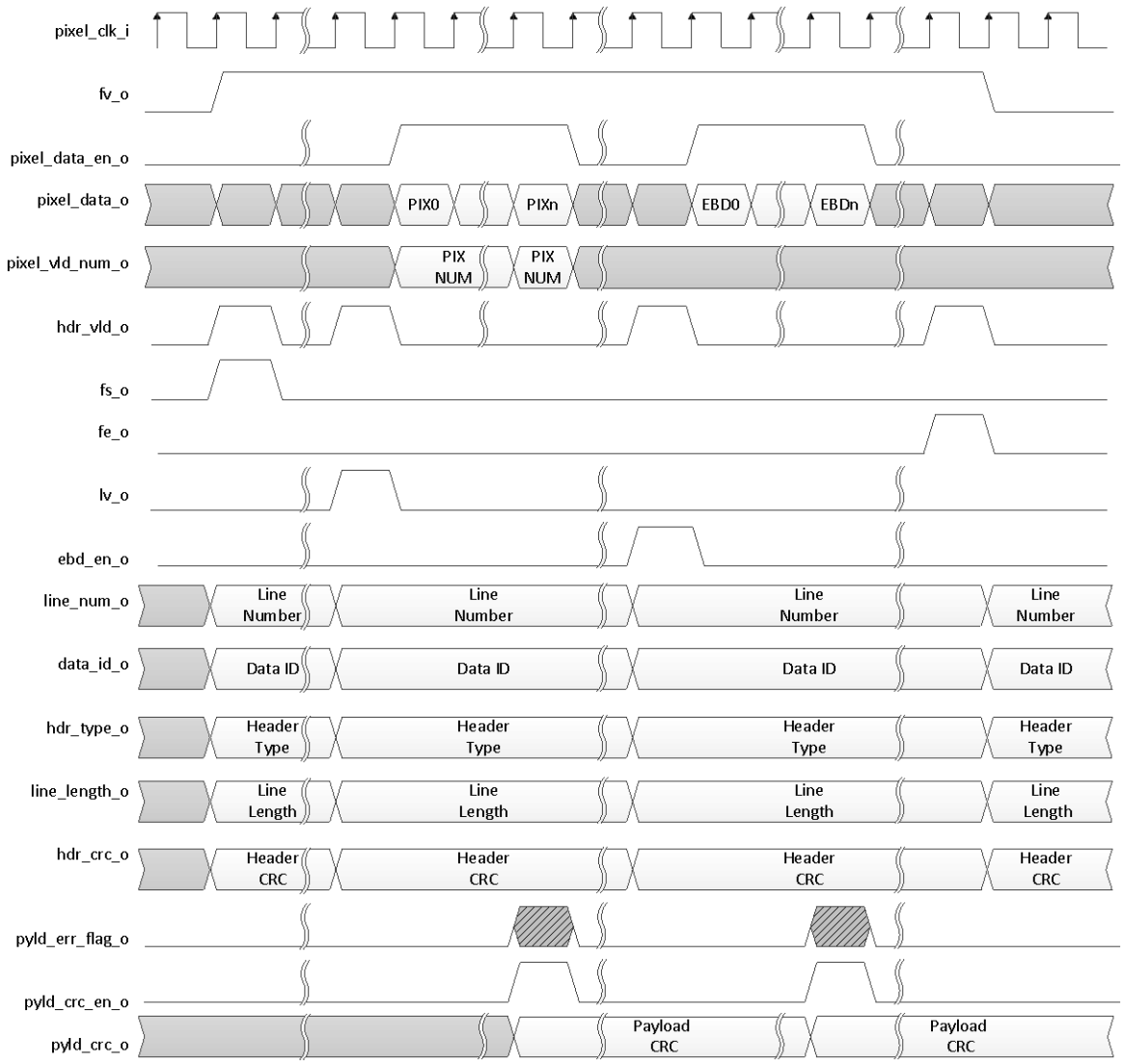



**Figure 2.10. SLVS-EC Rx Native Interface Output Timing Diagram**

## 2.8.2. SLVS-EC Rx with AXI4-Stream Transmitter Interface

Figure 2.11 shows the timing diagram for SLVS-EC Rx with the AXI4-Stream interface. The AXI4-Stream Transmitter interface does not support back-pressure and assumes that m_tready_i is always high. A transfer occurs when m_tvalid_o is asserted.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

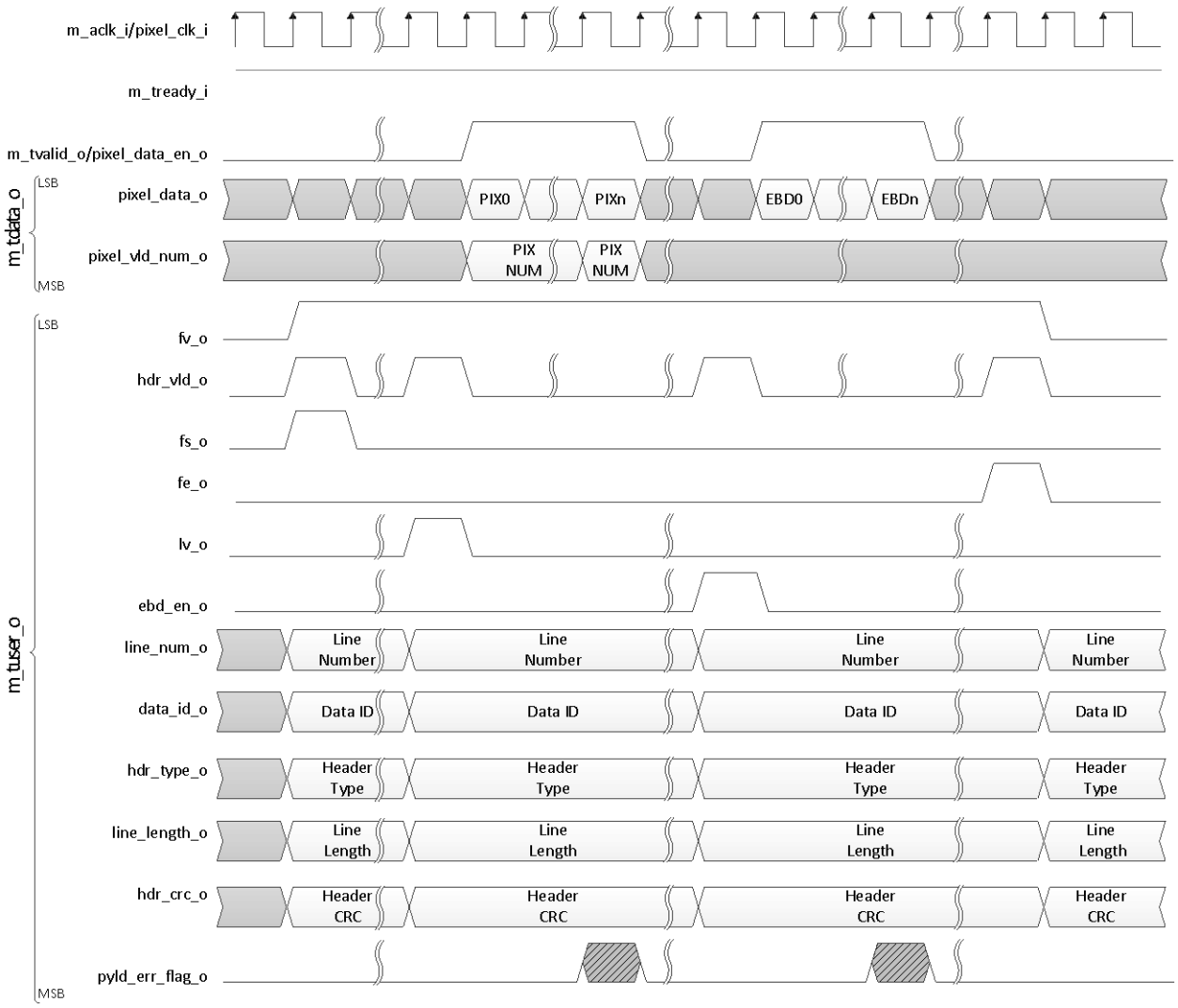



**Figure 2.11. SLVS-EC Rx AXI4-Stream Interface Timing Diagram**

## 2.8.3. SLVS-EC Receiver UVSI

Figure 2.12 shows the timing diagram for video-frame reception over UVSI when axis_vid_tready_i remains asserted throughout an Rx transaction.

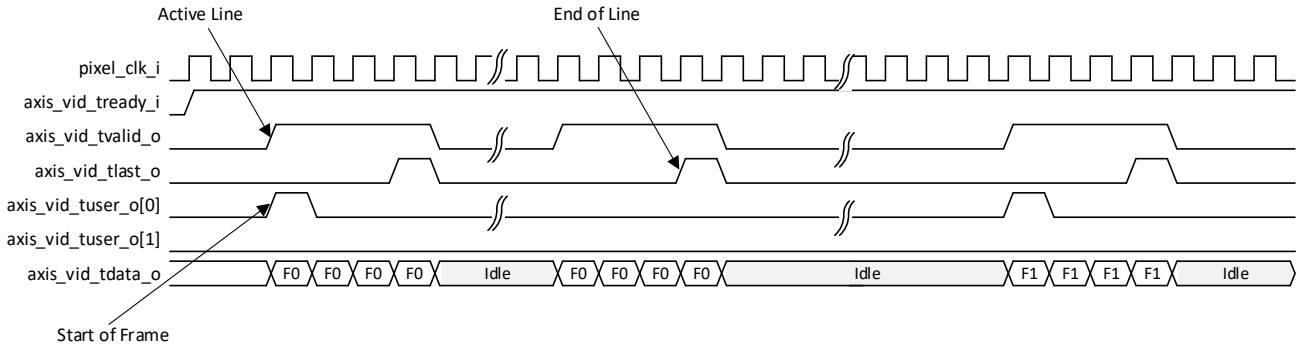



**Figure 2.12. UVSI Rx Timing Diagram without TREADY Deassertion**

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

When axis_vid_tready_i is deasserted (low) during an active packet transfer, as shown in Figure 2.13, pixels are stored in a pixel buffer. If axis_vid_tready_i remains low for a duration greater than or equal to the buffer depth (fixed at 128 words), the buffer may overflow, which can cause functional errors.

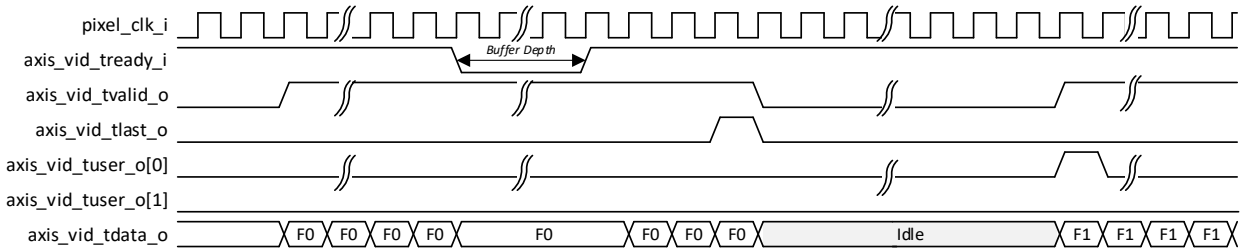



**Figure 2.13. UVSI Rx Timing Diagram with TREADY Deassertion**

### 2.8.3.1. TDATA Mapping

Key definitions used in TDATA mapping are as follows:

- Colors per Pixel (CPP): Number of color components per pixel. SLVS-EC UVSI supports only the RAW format; therefore, CPP = 1.
- Bits per Color (BPC): Configured bit width for each color component (set through the *Data Type (RAW Data Format)* IP parameter: 8, 10, 12, 14, or 16).
- Bits per Pixel (BPP): Total number of bits per pixel, where BPP = CPP × BPC.
- TDATA Width per Pixel: Width of the 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, where TDATA_WIDTH = TDATA Width per Pixel × PPC (*Number of Output Pixels per Pixel Clock (UVSI)*).
- Padding Bits: Any bits within the pixel TDATA slot above its BPC data are unused. For SLVS-EC UVSI, these padding bits are tied to 0.

The TDATA mapping for each data type and the sample configurations are shown in the table and figures below.

**Table 2.4. TDATA Mapping per Data Type**

| Data Type | CPP | BPC | BPP | TDATA Width per Pixel |
|---|---|---|---|---|
| RAW8 | 1 | 8 | 1 × 8 = 8 bits | Ceil(8/8) × 8 = 8 bits<br>No padding because BPP is already byte-aligned. |
| RAW10 | 1 | 10 | 1 × 10 = 10 bits | Ceil(10/8) × 8 = 16 bits |
| RAW12 | 1 | 12 | 1 × 12 = 12 bits | Ceil(12/8) × 8 = 16 bits |
| RAW14 | 1 | 14 | 1 × 14 = 14 bits | Ceil(14/8) × 8 = 16 bits |
| RAW16 | 1 | 16 | 1 × 16 = 16 bits | Ceil(16/8) × 8 = 16 bits<br>No padding because BPP is already byte-aligned. |

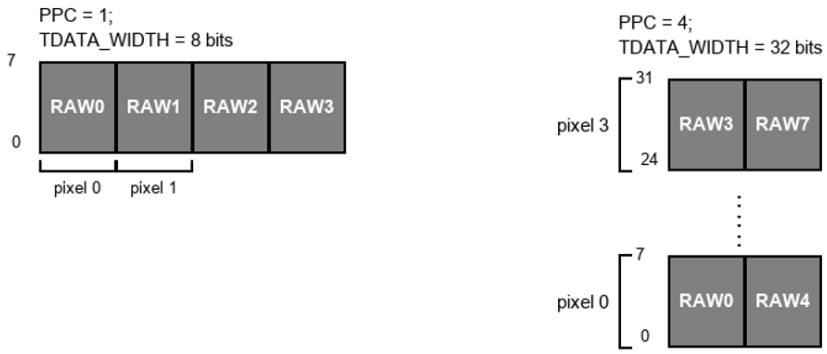



**Figure 2.14. TDATA Pixel Mapping for Sample RAW8 Configuration**

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

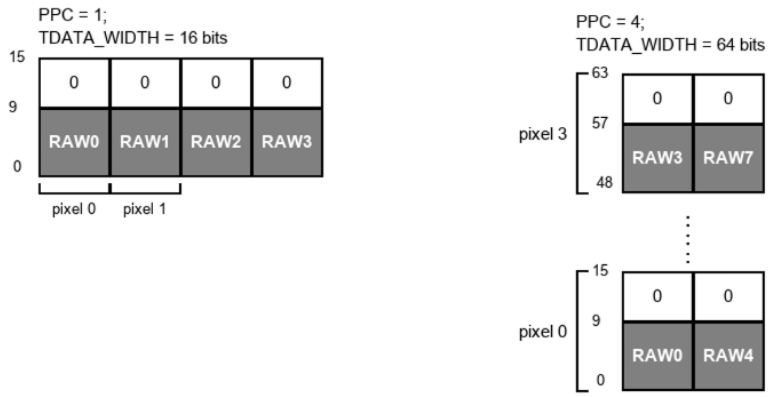



**Figure 2.15. TDATA Pixel Mapping for Sample RAW10 Configuration**

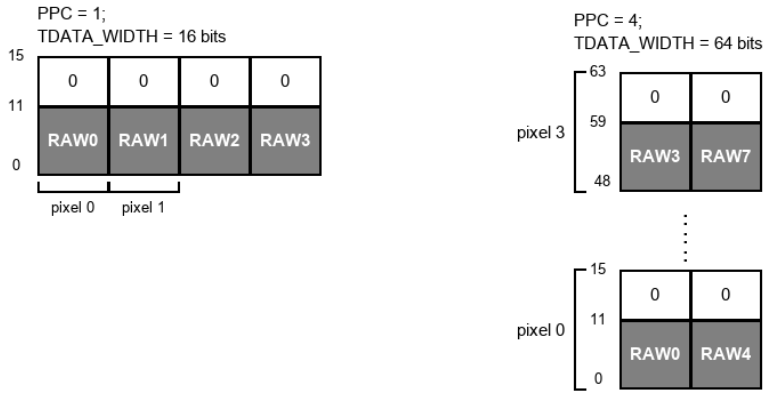



**Figure 2.16. TDATA Pixel Mapping for Sample RAW12 Configuration**

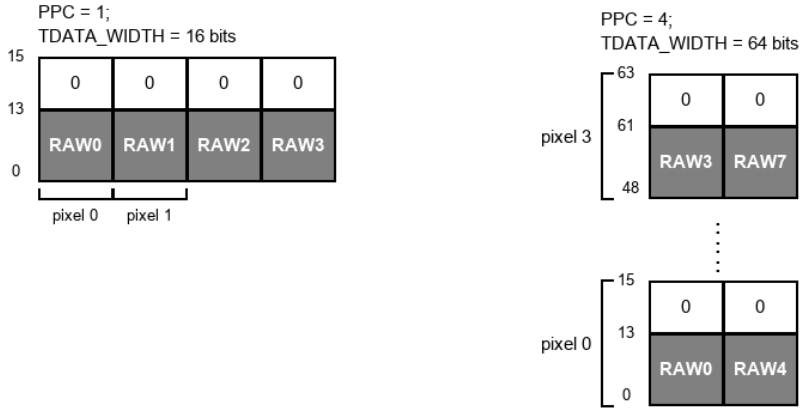



**Figure 2.17. TDATA Pixel Mapping for Sample RAW14 Configuration**

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

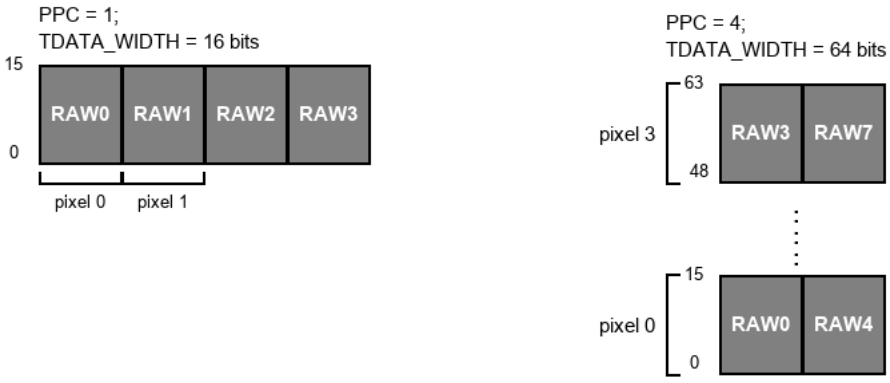



**Figure 2.18. TDATA Pixel Mapping for Sample RAW16 Configuration**

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# 3. IP Parameter Description

The configurable attributes of the SLVS-EC Receiver IP are shown in the following table. You can configure the IP by setting the attributes accordingly in the IP Catalog Module/IP 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 |
|---|---|---|
| **General** | | |
| Number of Rx Lanes | 1, 2, 4, 6, **8** | Number of SLVS-EC Rx lanes. |
| PCS Lane ID | **AUTO,** 0, 1, 2, 3, 4, 5, 6, 7 | Specifies the location of the first lane of the PCS instance. Refer to the MPCS Module User Guide (FPGA-IPUG-02118) for the rules in selecting the Lane ID. |
| Data Type (RAW Data Format) | 8, 10, 12, 14, **16** | Number of Bits per Pixel in RAW Data Format. |
| **Data Interface Type** | | |
| Enable Native Interface | **Checked**, Unchecked | Select to use the Native interface as outputs. |
| Enable AXI4-Stream Transmitter Interface | Checked, **Unchecked** | Select to use the AXI4-Stream Transmitter interface as outputs. |
| Enable Unified Video Streaming Transmitter Interface | Checked, **Unchecked** | Select to use the UVSI Transmitter interface as outputs. |
| **Data Rate** | | |
| Baud Grade | 1, 2, **3** | Baud grade setting:<br>• 3: Baud Rate = up to 5,000 Mbps<br>• 2: Baud Rate = up to 2,500 Mbps<br>• 1: Baud Rate = up to 1,250 Mbps |
| Reference Clock Frequency (MHz) | 74.25–**125** | Reference clock frequency of the MPCS soft IP (refclk_n_i/refclk_p_i). The MPCS PLL settings for each baud grade are pre-configured. MPCS/SERDES supports only a limited range of reference clock frequencies. For details, refer to the CertusPro-NX SERDES/PCS User Guide (FPGA-TN-02245). |
| MPCS Bit Rate (Mbps) | Calculated | MPCS data rate per lane. Calculated values depend on the *Reference Clock* and *Baud Grade*. |
| MPCS Effective Data Rate (Mbps) | Calculated | Total effective MPCS output data rate. Calculated values depend on the *Reference Clock*, *Baud Grade*, and *Number of Rx Lanes*. |
| **Pixel Interface** | | |
| Number of Output Pixels per Pixel Clock | Calculated | Fixed value of output pixels per pixel clock depending on the *Data Type*, MPCS Data Bytes, and *Number of Rx Lanes*. MPCS Data Bytes varies depending on the *Baud Grade* setting:<br>• 1 byte if Baud Grade = 1<br>• 2 bytes if Baud Grade = 2<br>• 4 bytes if Baud Grade = 3<br><br>For RAW16 Data Type:<br>If Number of Rx Lanes = 1 and Baud Grade = 1, then the Number of Output Pixels per Pixel Clock = 1.<br>Otherwise, the Number of Output Pixels per Pixel Clock = (MPCS data bytes × Number of Rx Lanes) / 2. |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

| Attribute | Selectable Values | Description |
|---|---|---|
| | | For other Data Types:<br>The Number of Output Pixels per Pixel Clock = MPCS data bytes × Number of Rx Lanes. |
| Pixel Clock Frequency (MHz) | 10–250, **125** | Input clock frequency for the pixel clock domain. |
| Effective Output Data Rate (Mbps) | Calculated | Effective output data rate in the pixel clock domain. |
| **FIFO** | | |
| Manual Adjust | Checked, **Unchecked** | Enable to manually adjust *FIFO Depth* and *FIFO Read Delay* settings. Auto-computations of GUI *FIFO Depth* and *FIFO Read Delay* attributes do not consider the pad codes inserted in the packet data. Adjust manually to avoid FIFO Empty and Full states during packet transfer. Refer to the Byte-to-Pixel Converter section to get the calculation for *FIFO Depth* and *FIFO Read Delay*. |
| Pixel Count | 1–16,384, **1920** | Number of pixels per line. |
| FIFO Depth | 2–65,536, **16** | Set Byte-to-Pixel FIFO depth. |
| FIFO Read Delay | 1–65,536, **1** | Set Byte-to-Pixel delay cycles for the FIFO read threshold. |
| **Header Analysis** | | |
| Enable Header Analysis | **Checked**, Unchecked | Packet header information is parsed if enabled and is available in the output. |
| **Miscellaneous** | | |
| Enable Miscellaneous Signals | Checked, **Unchecked** | Miscellaneous signals are available in the output if enabled. |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# 4. Signal Description

This section describes the SLVS-EC Receiver IP ports.

## 4.1. Clock and Reset

**Table 4.1. Clock and Reset Ports**

| Port | Type | Description |
|---|---|---|
| reset_n_i | Input | Active-low asynchronous reset. |
| refclk_n_i | Input | Differential Reference Clock (CLK–). |
| refclk_p_i | Input | Differential Reference Clock (CLK+). |
| mpcs_clkin_i | Input | Clock source for PMA inside the MPCS soft IP. Drives all calibration logic inside the PMA Controller and can be driven using any frequency as long as it is compliant with the CertusPro-NX SERDES/PCS User Guide (FPGA-TN-02245). |
| mpcs_pwrdn_i[1:0] | Input | PHY power down:<br>• 2'b00: Operational<br>• 2'b10: Low-power state. Rx CDRPLL is powered down. The PHY starts in this low-power mode to perform calibration.<br>• 2'b11: Deep low-power state. Tx driver is in electrical Idle II (static low). All PLLs are powered down and the clocks are shut down. |

## 4.2. Primary I/O

**Table 4.2. Primary I/O Ports**

| Port | Type | Description |
|---|---|---|
| d_p_i[N-1:0] | Input | Differential Receive Serial Interface, Rx+, N: number of lanes. |
| d_n_i[N-1:0] | Input | Differential Receive Serial Interface, Rx–, N: number of lanes. |

## 4.3. Native Interface

**Table 4.3. Native Interface Ports**

| Port | Type | Description |
|---|---|---|
| pixel_clk_i | Input | Pixel clock input that clocks the byte-to-pixel conversion logic in the pixel domain for the Native Interface.<br>When *Enabled Unified Video Streaming Interface* is selected, it clocks the UVSI byte-to-pixel conversion logic and the AXI4-Stream video output logic.<br>When UVSI is selected, *pixel_clk_i* must satisfy the following requirement:<br>• Pixel Clock (MHz) ≥ (MPCS Effective Data Rate (Mbps) / *Byte-to-Pixel Data Width* (Bits))<br>For more information, refer to Table 4.4. |
| fv_o | Output | Frame valid output for active lines. |
| pixel_data_en_o | Output | Pixel data enable. |
| pixel_data_o[PIX_WIDTH-1:0] | Output | Pixel data output in LSB first data order.<br>PIX_WIDTH = Bits per Pixel (data type) × Number of Output Pixels per Pixel Clock |
| pixel_vld_num_o[7:0] | Output | Indicates valid number of output pixels per cycle. |

**Note:**

- Native interface ports are available if *Enable Native Interface* is enabled.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

**Table 4.4. Byte-to-Pixel Data Width by Baud Grade and Number of Rx Lanes**

| Baud Grade | Number of Rx Lanes | Byte-to-Pixel Data Width (Bits) |
|---|---|---|
| 1 | 1 | 32 |
| 1 | 2 | 32 |
| 1 | 4 | 32 |
| 1 | 6 | 64 |
| 1 | 8 | 64 |
| 2 | 1 | 32 |
| 2 | 2 | 32 |
| 2 | 4 | 64 |
| 2 | 6 | 128 |
| 2 | 8 | 128 |
| 3 | 1 | 32 |
| 3 | 2 | 64 |
| 3 | 4 | 128 |
| 3 | 6 | 256 |
| 3 | 8 (default) | 256 |

## 4.4. Packet Header Information

**Table 4.5. Packet Header Information Ports**

| Port | Type | Description |
|---|---|---|
| lv_o | Output | Indicates whether the line is valid or invalid. |
| fs_o | Output | Indicates the beginning of a frame. |
| fe_o | Output | Indicates the end of a frame. |
| line_num_o[12:0] | Output | Indicates the line number. |
| data_id_o[3:0] | Output | Data ID for multiple stream transfer. |
| hdr_type_o[2:0] | Output | 3-bit Header Type field from the received packet header is valid when *hdr_vld_o* is asserted. Only 3'b001 is interpreted internally and indicates that a valid embedded line-length field is present. All other codes are passed through unmodified. For more information, refer to the documentation provided by the specific SLVS-EC sensor or transmitter vendor. |
| line_length_o[15:0] | Output | Payload word count per line (hdr_type_o = 3'b001). |
| hdr_crc_o[15:0] | Output | Header CRC. |
| ebd_en_o | Output | Indicates the line with the embedded data. |
| hdr_vld_o | Output | Indicates the valid header information. |

**Note:**

- Packet header information ports are available if *Enable Header Analysis* is enabled.

## 4.5. Control and Status

**Table 4.6. Control and Status Ports**

| Port | Type | Description |
|---|---|---|
| mpcs_ready_o | Output | Active-high ready signal indicating that the PHY has completed the calibration sequence for each specific lane. MPCS is ready to receive the training sequence. |
| setup_i | Input | Active-high setup signal from the user interface to commence start-up. |
| ready_o | Output | Active-high ready signal indicating that Rx is ready for packet reception. Training sequence is complete. |
| pyld_err_flag_o | Output | Payload error detected. |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

## 4.6. Miscellaneous

**Table 4.7. Miscellaneous Ports**

| Port | Type | Description |
|---|---|---|
| mpcs_rx_clk_o | Output | Output clock from the MPCS soft IP (MPCS Rx clock). |
| pyld_crc_o | Output | Packet footer information. Payload CRC. |
| pyld_crc_en_o | Output | Valid payload CRC. |

**Note:**

- Miscellaneous ports are available if *Enable Miscellaneous Signals* is enabled.

## 4.7. AXI4-Stream Transmitter Interface

**Table 4.8. AXI4-Stream Transmitter Interface Ports**

| Port | Type | Description |
|---|---|---|
| m_aclk_i | Input | Clock for the pixel domain logic. |
| m_tvalid_o | Output | TVALID indicates that the transmitter is driving a valid transfer. A transfer takes place when both TVALID and TREADY are asserted. |
| m_tready_i | Input | The AXI4-Stream Transmitter interface does not accept back-pressure and assumes an always-high m_tready_i. |
| m_tdata_o[PIX_WIDTH+8-1:0] | Output | Pixel Output Data.<br>PIX_WIDTH = Bits per Pixel (data type) × Number of Output Pixels per Pixel Clock |
| m_tuser_o | Output | Sideband information transmitted alongside the data stream. |

**Note:**

- AXI4-Stream Transmitter interface ports are available if *Enable AXI4-Stream Transmitter Interface* is enabled.

## 4.8. Unified Video Streaming Interface

**Table 4.9. 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, this input causes the internal FIFO of the module to buffer pixel data (up to its fixed depth); the byte-to-pixel converter itself continues to run and does not receive back-pressure. |
| 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[UVSI_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 Output Pixels per Pixel Clock (UVSI)* > 1. Valid only when axis_vid_tvalid_o is asserted. The minimum UVSI_WIDTH width is 1 byte and may include padding bits for alignment to byte boundaries. Refer to the SLVS-EC Receiver UVSI section for details. |
| axis_vid_tuser_o[1:0] | Output | AXI4-Stream video user sideband signals. Provides frame and line timing information.<br>axis_vid_tuser_o[0]: Start of Frame (SOF) – Asserts for the first pixel of each frame.<br>axis_vid_tuser_o[1]: Interlaced F1 – Currently unused and is tied 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. |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

## 4.9. APB Interface

**Table 4.10. APB Interface Ports**

| Port | Type | Description |
|---|---|---|
| apb_pclk_i | Input | APB clock. Operates at 125 MHz. |
| apb_preset_n_i | Input | Reset (active low). |
| apb_psel_i | Input | Select signal. Indicates that the completer device is selected and a data transfer is required. |
| apb_paddr_i | Input | Address signal. |
| apb_pwdata_i | Input | Write data signal. |
| apb_pwrite_i | Input | Direction signal. Write = 1, Read = 0. |
| apb_penable_i | Input | Enable signal. Indicates the second and subsequent cycles of an APB transfer. |
| apb_pready_o | Output | Ready signal. Indicates transfer completion. The completer device uses this signal to extend an APB transfer. |
| apb_prdata_o | Output | Read data signal. |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# 5. Register Description

The read and write behavior of each register is defined by its access type, as defined in Table 5.1.

**Table 5.1. Access Type Definition**

| Access Type | Behavior on Read Access | Behavior on Write Access |
|---|---|---|
| RO | Returns register value | Ignores write access |
| WO | Returns 0 | Updates register value |
| RW | Returns register value | Updates register value |
| RW1C | Returns register value | Writing 1'b1 on the register bit clears the bit to 1'b0. Writing 1'b0 on the register bit is ignored. |
| RSVD | Returns 0 | Ignores write access |

**Table 5.2. Summary of SLVS-EC Receiver Registers**

| Offset | Register Name | Access | Default Value | Register Description | | |
|---|---|---|---|---|---|---|
| | | | | Bit | Field | Field Description |
| **Attribute Registers** | | | | | | |
| 0x00 | SYNC_SYMBOL | RW | D.10.5 | [7:0] | sync_symbol | Set 8-bit symbol following the comma symbol (K28.5) within the Sync Code. |
| | | | | [31:8] | RSVD | Reserved bits. |
| 0x04 | DESKEW_SYMBOL | RW | D.00.3 | [7:0] | deskew_symbol | Set 8-bit symbol following the comma symbol (K28.5) within the Deskew Code. |
| | | | | [31:8] | RSVD | Reserved bits. |
| 0x0C | STDBY_SYMBOL | RW | D.03.0 | [7:0] | stdby_symbol | Set 8-bit symbol following the comma symbol (K28.5) within the Standby Code. |
| | | | | [31:8] | RSVD | Reserved bits. |
| 0x10 | SYNC_LENGTH[1] | RW | 24'd255 | [23:0] | sync_length | Set the Sync Code transfer times. |
| | | | | [31:24] | RSVD | Reserved bits. |
| 0x14 | DESKEW_LENGTH[1] | RW | 8'd16 | [7:0] | deskew_length | Set the Deskew Code transfer times. |
| | | | | [31:8] | RSVD | Reserved bits. |
| 0x18 | DESKEW_INTERVAL | RW | 8'd16 | [7:0] | deskew_interval | Set the Deskew Code transmission interval (number of times the Idle Code is sent between Deskew Codes). |
| | | | | [31:8] | RSVD | Reserved bits. |
| 0x1C | STDBY_LENGTH | RW | 8'd16 | [7:0] | stdby_length | Set the Standby Code transfer times. |
| | | | | [31:8] | RSVD | Reserved bits. |
| 0x20 | INITIAL_LENGTH | RW | 16'd2047 | [15:0] | initial_length | Set the differential low output period during mode change and initialization. |
| | | | | [31:16] | RSVD | Reserved bits. |
| 0x24 | IDLE_CODE | RW | D.00.0 | [7:0] | idle_code | Set the symbol for the Idle Code. |
| | | | | [31:8] | RSVD | Reserved bits. |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

| Offset | Register Name | Access | Default Value | Register Description | | |
|---|---|---|---|---|---|---|
| | | | | Bit | Field | Field Description |
| **Configuration Registers[2]** | | | | | | |
| 0x28 | LANE_WIDTH | RW | {depends on the *Number of Rx Lanes* attribute} | [3:0] | lane_width | Set the number of SLVS-EC Rx lanes.<br>1: 1 lane<br>2: 2 lanes<br>4: 4 lanes<br>6: 6 lanes<br>8: 8 lanes<br>Others: Setting prohibited |
| | | | | [31:4] | RSVD | Reserved bits. |
| 0x2C | LINE_LENGTH | RW | {depends on the *Pixel Count* attribute} | [15:0] | line_length | Set the number of pixels per line. |
| | | | | [31:16] | RSVD | Reserved bits. |
| 0x30 | PIXEL_BIT | RW | {depends on the *Data Type (RAW Data Format)* attribute} | [4:0] | pixel_bit | Set bits per pixel based on the data type.<br>8: RAW8<br>10: RAW10<br>12: RAW12<br>14: RAW14<br>16: RAW16<br>Others: Setting prohibited |
| | | | | [31:5] | RSVD | Reserved bits. |
| 0x34 | BAUD_GRADE | RW | {depends on the *Baud Grade* attribute} | [1:0] | baud_grade | Set grade of PHY baud rate.<br>1: Set Baud Grade 1<br>2: Set Baud Grade 2<br>3: Set Baud Grade 3<br>Others: Setting prohibited |
| | | | | [31:2] | RSVD | Reserved bits. |
| 0x38 | PACKET_FORMAT[3] | RW | {29'd0,2'd0,1'd1} | [0] | crc_en | Set Payload CRC or packet footer.<br>0: Disables CRC<br>1: Enables CRC |
| | | | | [2:1] | ecc_opt | Set Payload ECC Option.<br>0: Disables ECC<br>1: Enables ECC with 2 parity bytes, 18~226 block length in bytes<br>2: Enables ECC with 4 parity bytes, 20~228 block length in bytes<br>Others: Setting prohibited |
| | | | | [31:3] | RSVD | Reserved bits. |
| 0x3C | FIFO_DELAY | RW | {depends on the *FIFO Read Delay* attribute} | [15:0] | fifo_delay | Set delay cycles for reading the FIFO. |
| | | | | [31:16] | RSVD | Reserved bits. |

**Notes:**

1. This register does not imply how many SYNC/DESKEW packets are needed to achieve alignment; rather, it states the number of packet transfers, and does not necessarily mean that the IP can assure alignment using the programmed value. The transmitter needs to send enough codes, within the SLVS-EC Receiver requirement, to achieve alignment.
2. Reconfiguration of IP through registers is only supported if the new target configuration is smaller/slower than initially set (that is, BAUD_GRADE==3 to BAUD_GRADE==2; LANE_WIDTH==8 to LANE_WIDTH==6/4/2/1; and so on).

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

3. Packet footer (CRC) option and ECC for payload data cannot be enabled simultaneously. ECC is currently not supported; access to [2:1] is RO.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# 6. Designing with the IP

This section describes how to generate the IP core using the Lattice Radiant software and how to run simulation and synthesis. For more information about 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.

## 6.1. Generating and Instantiating the IP

You can use the Lattice Radiant software to generate IP modules and integrate them into the device architecture. The steps below describe how to generate the SLVS-EC Receiver IP in the Lattice Radiant software.

To generate the SLVS-EC Receiver IP:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click **SLVS-EC Receiver** under **IP, Audio_Video_And_Image_Processing** category. The **Module/IP Block Wizard** opens as shown in Figure 6.1. Enter values in the **Component name** and the **Create in** fields and click **Next**.

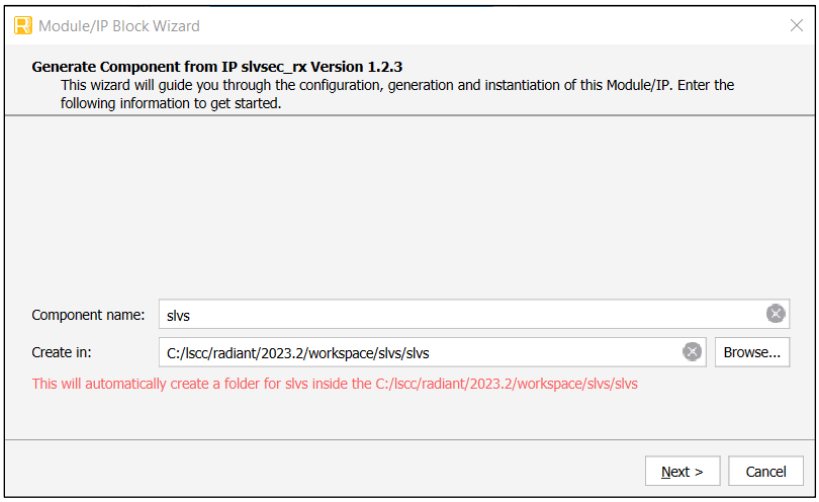



**Figure 6.1. Module/IP Block Wizard**

3. In the next **Module/IP Block Wizard** window, customize the selected SLVS-EC Receiver IP using drop-down lists and check boxes. Figure 6.2 shows an example configuration of the SLVS-EC Receiver IP. For details on the configuration options, refer to the IP Parameter Description section.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

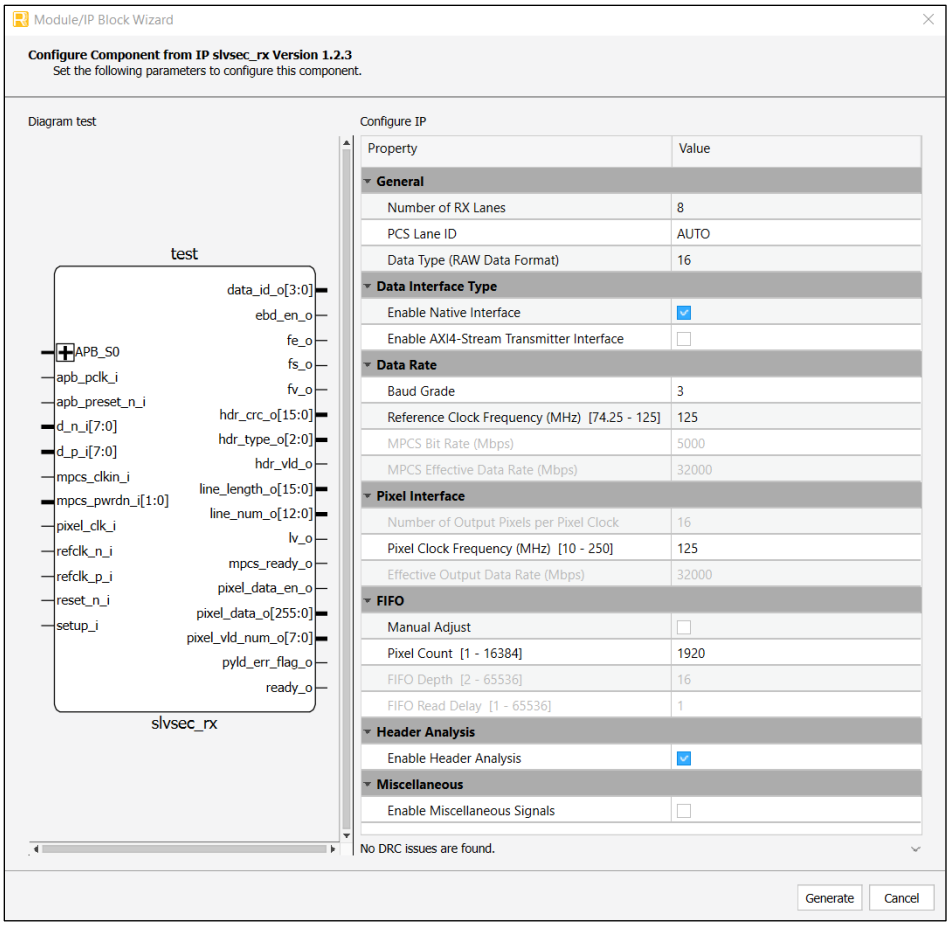



**Figure 6.2. IP Configuration**

4. Click **Generate**. The **Check Generated Result** dialog box opens, showing design block messages and results as shown in Figure 6.3.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

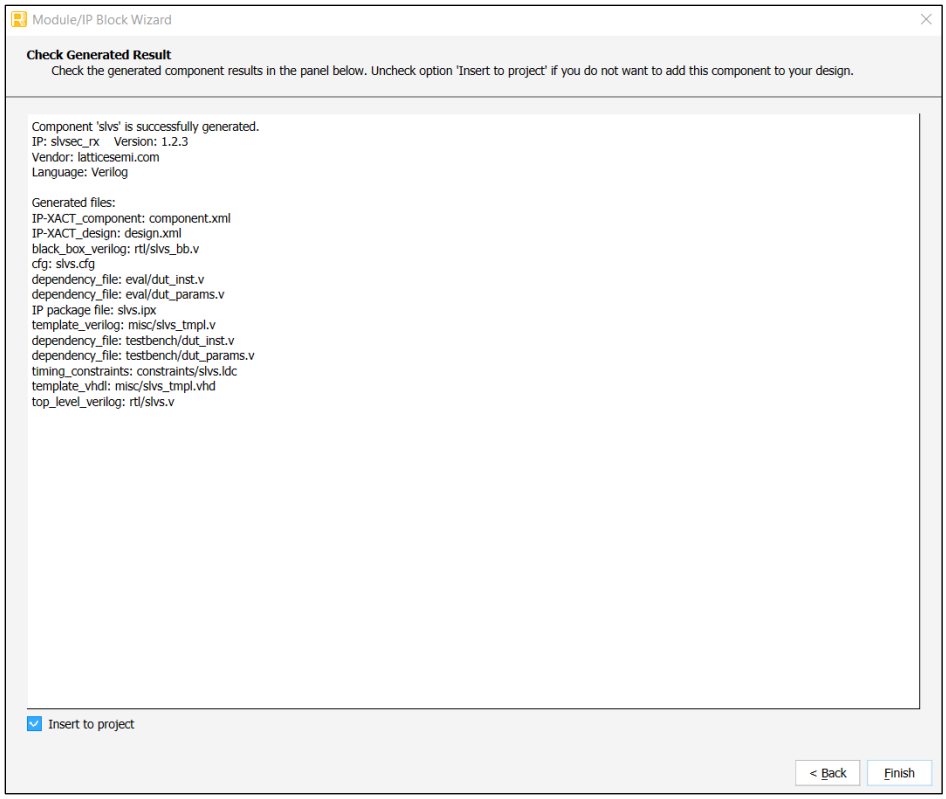



**Figure 6.3. Check Generated Result**

5. Click **Finish**. All the generated files are placed under the directory paths in the **Create in** and the **Component name** fields shown in Figure 6.1.

### 6.1.1. Generated Files and File Structure

The generated SLVS-EC Receiver 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 top-level reference as the starting template for the top-level of the complete design. The generated files are listed in Table 6.1.

**Table 6.1. Generated File List**

| Attribute | Description |
|---|---|
| <Component name>.ipx | This file contains the information on the files associated with the generated IP. |
| <Component name>.cfg | This file 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 IP-XACT 2014 format. |
| rtl/<Component name>.v | This file provides an example RTL top file that instantiates the module. |
| rtl/<Component name>_bb.v | This file provides the synthesis closed-box. |
| misc/<Component name>_tmpl.v<br>misc/<Component name>_tmpl.vhd | These files provide instance templates for the module. |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

## 6.2. Design Implementation

Completing your design includes additional steps to specify analog properties, pin assignments, and timing and physical design 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 design 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.

## 6.3. Timing and Physical Design Constraints

You need to provide appropriate timing and physical design constraints to ensure that your design meets the desired performance goals on the FPGA device. The timing constraints are based on the clock frequency used and the interfaces enabled for the IP. The timing constraints for the IP are defined in the relevant constraint file:

```
<IP_Instance_Path>/<IP_Instance_Name>/constraints/<IP_Instance_Name>.ldc.
```

The example below shows the IP timing constraint *.ldc* file generated for the SLVS-EC Receiver IP:

```
set PIX_CLK_PERIOD [expr {double(round(1000000/$PIX_CLK))/1000}]

create_clock -name {refclk_p_i}   -period 8 [get_ports refclk_p_i]
create_clock -name {refclk_n_i}   -period 8 [get_ports refclk_n_i]
create_clock -name {apb_pclk_i}   -period 8 [get_ports apb_pclk_i]
create_clock -name {mpcs_clkin_i} -period 8 [get_ports mpcs_clkin_i]

if {$NATIVE_INTF == "ENABLED" || $UVSI_INTF == "ENABLED"} {
create_clock -name {pixel_clk_i}  -period $PIX_CLK_PERIOD [get_ports pixel_clk_i]
}

if {$AXI4_INTF == "ENABLED"} {
create_clock -name {m_aclk_i}     -period $PIX_CLK_PERIOD [get_ports m_aclk_i]
}
```

The constraint file has been verified during IP evaluation with the IP instantiated directly in the top-level module. To use it, copy the content of the *.ldc* file shown above into your top-level design constraints (*.pdc*) for post-synthesis implementation, and then add the additional constraints described below.

The clocks shown in the timing constraint file are generated and ready to use as is. However, you must complete several additional constraints based on your design instance hierarchy.

For timing closure, add the following timing constraints to your top-level *.pdc* file:

- Copy the active *create_clock* statement shown in the timing constraint file as is. No modification is required.
- Create the recovered MPCS Rx lane clock (*mpcs_rx_clk_w*) using the commented template in the *.ldc* file. Replace the example instance name *lscc_slvsec_rx_inst* with the actual hierarchical path of the IP instance in your design. Check your netlist or PAR timing report for the correct path. Then copy the *set_clock_groups* statement immediately below it in the *.ldc* file as is.
- Copy and uncomment the remaining constraint templates for the IP internal configuration and status false paths (*stdby**, *cfg_** *attr_**, and *mpcs-ready**) and the LMMI handshake multicycle-path exceptions. Replace *lscc_slvsec_rx_inst* with the actual instance path in each constraint.

Only uncomment the optional false-path templates (APB/pixel/MPCS clock) if the PAR/STA report still shows unresolved cross-domain violations after you add the asynchronous clock groups. Do not enable these speculatively.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

Note that the *.ldc* file also includes the following constraint:

```
set_false_path -from [get_nets -hierarchical *reset*]
```

Do not copy this line as is. The unscoped **reset** wildcard matches any net containing **reset** anywhere in the compiled design, not just within the IP, and can unintentionally exempt reset paths elsewhere in the timing analysis. Replace it with an instance-scoped version instead:

```
set_false_path -from [get_nets -hierarchical {<IP_Instance_Path>/*reset*}]
```

Refer to the Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059) for additional information on constraining your design.

## 6.4. Specifying the Strategy

The Lattice Radiant software provides two predefined strategies: Area and Timing. It 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.

## 6.5. Running Functional Simulation

You can run functional simulation after the IP is generated.

To run functional simulation:

1. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in Figure 6.4.

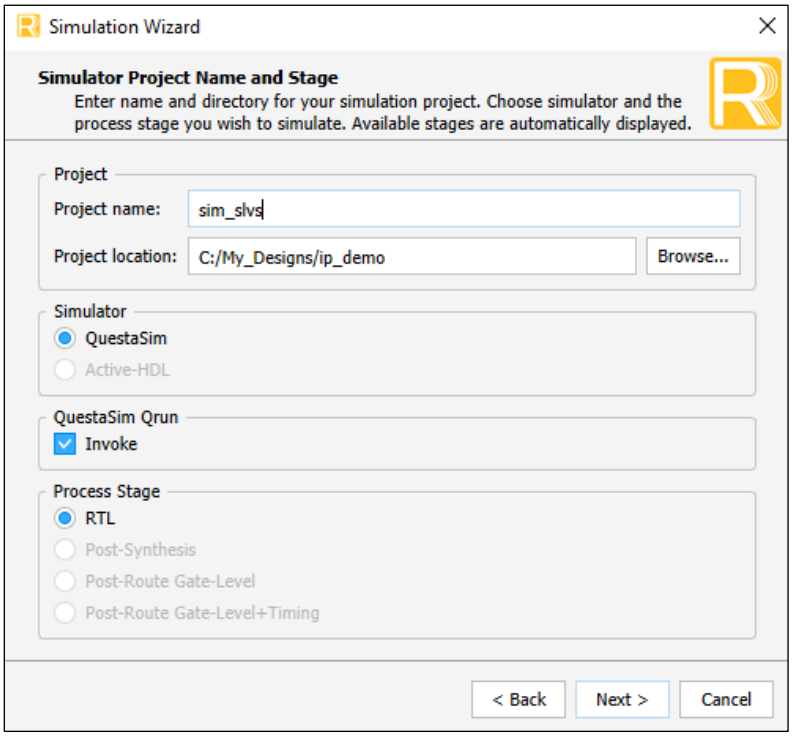



**Figure 6.4. Simulation Wizard**

2. Click **Next** to open the **Add and Reorder Source** window as shown in Figure 6.5.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

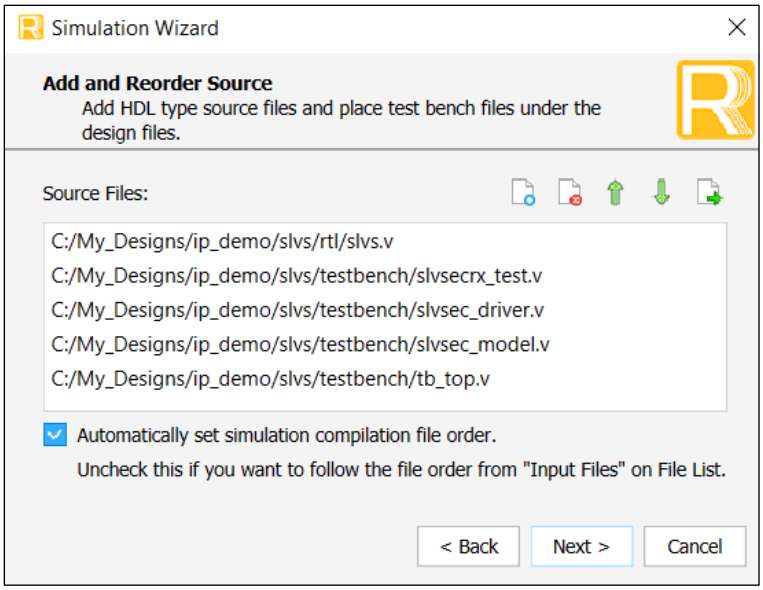



**Figure 6.5. Add and Reorder Source**

3. Click **Next**. The **Parse HDL files for simulation** window is as shown in Figure 6.6. Ensure the **Simulation Top Module** is set to *tb_top*.

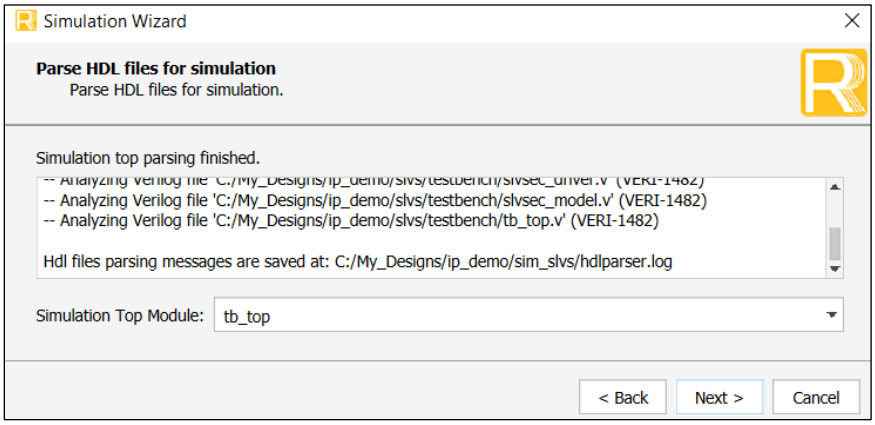



**Figure 6.6. Parse HDL files for simulation**

4. Click **Next**. The Summary window is as shown in Figure 6.7. Set Default Run to **0 ns** to ensure the simulation is complete.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

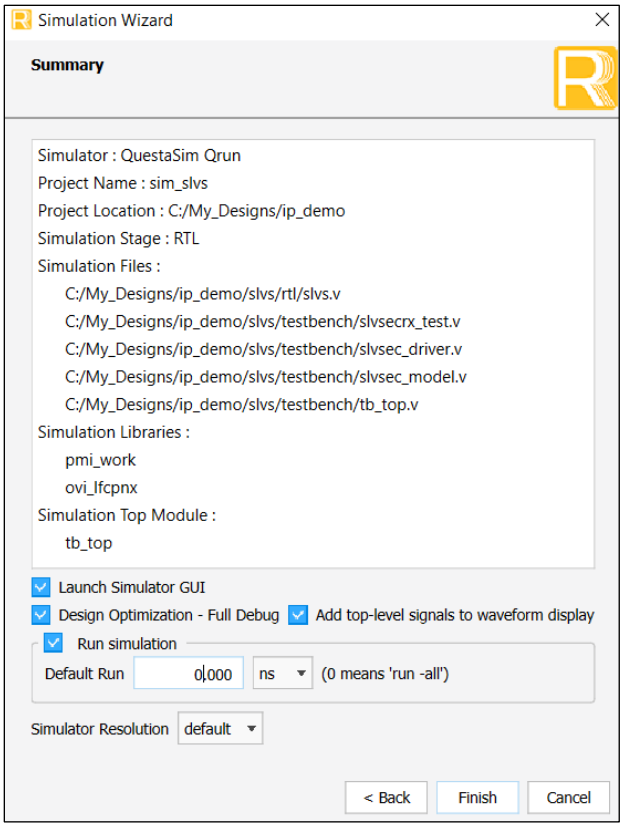



**Figure 6.7. Summary**

5. Click **Finish** to run the simulation.

The waveform in Figure 6.8 shows an example simulation result.

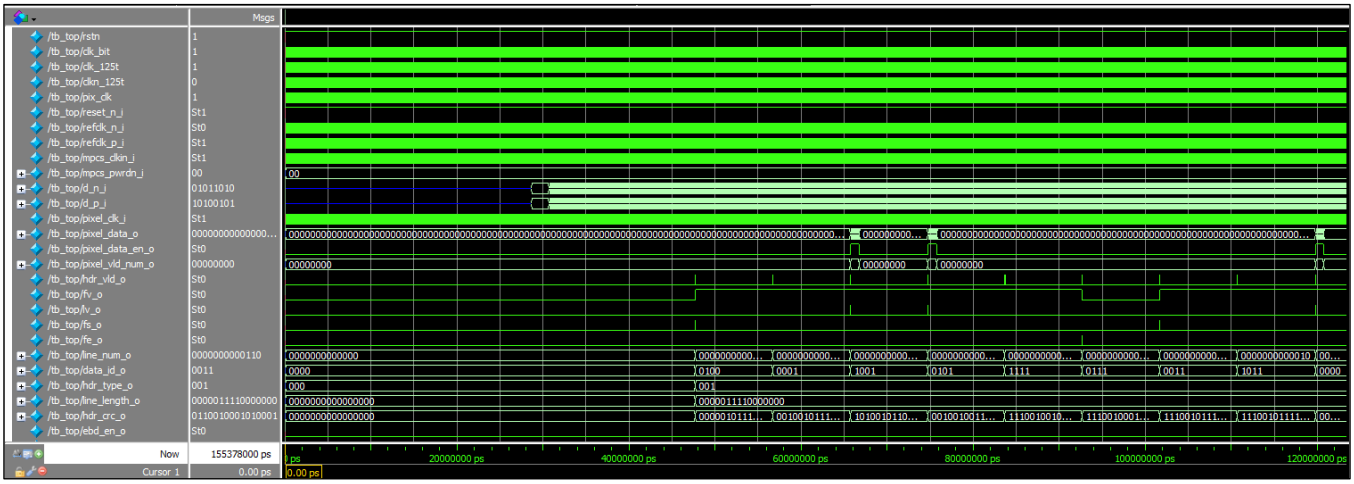


**Figure 6.8. Simulation Waveform**

After the simulation has been completed, the log with the *Simulation Done* message is printed as shown in Figure 6.9 below. Note that the IP testbench does not include a data comparator or checker. The testbench is set up only to send SLVS-EC-compliant stimulus to the IP interface ports.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

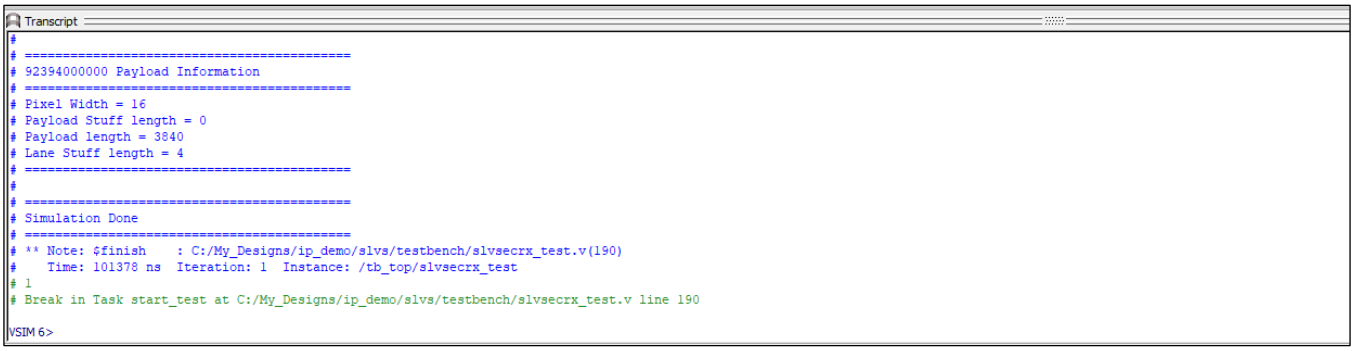



**Figure 6.9. Simulation Log on a Complete Run**

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# 7. Debugging

This section lists the suggested tools that you can use for debugging.

## 7.1. Debug Tools

You can use various tools to debug SLVS-EC Receiver IP design issues.

### 7.1.1. Reveal Analyzer

The Reveal Analyzer continuously monitors signals within the FPGA device for specific conditions, from simple to complex. 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 formats:

- 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 a hard or 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 device.
During debug cycles, this tool uses a divide-and-conquer method to narrow problem areas down to small functional blocks, so you can 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.

### 7.1.2. QuestaSim

Siemens EDA QuestaSim is an OEM simulator closely linked to the Lattice Radiant software environment. Use the simulator to perform functional verification of your design and IP. Write a testbench that provides input stimulus to the Device Under Test (DUT) and observe the output signals through the QuestaSim Waveform Viewer to verify the correctness of the IP or design. Lattice provides an IP testbench and other simulation files when you generate the IP in the Radiant software. You can use them to verify the behavior of the IP and as a reference during your debug activity. To run the IP simulation in QuestaSim, refer to the Running Functional Simulation section.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# Appendix A. Resource Utilization

Table A.1 shows a sample resource utilization of the SLVS-EC Receiver IP core on the LFCPNX-100-9BBG484C device using the Lattice Synthesis Engine of Lattice Radiant software 2024.1.0.

**Table A.1. Resource Utilization**

| Configuration | Clk Fmax (MHz)[1] | Registers | LUTs | Programmable I/O | DSP | sysMEM EBRs |
|---|---|---|---|---|---|---|
| 4 Lanes<br>RAW10<br>Baud Grade 3<br>FIFO Depth = 16<br>Native Interface | pixel_clk = 103.466 MHz<br>mpcs_clkin = 162.022 MHz | 9,940 | 29,030 | 10 | 6 | 24 |

**Note:**

1. Fmax is generated when the FPGA design contains only the SLVS-EC Receiver IP and the target frequency is 125 MHz. These values may be reduced when user logic is added to the FPGA design.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# Appendix B. Limitations

The unsupported features of the SLVS-EC Receiver IP are listed below:

- The IP does not support payload data Error Correction Code (ECC).
- The IP does not support the following system operations:
  - Standby
  - Mode Change with Standby
  - Interrupt
  - Embedded data
  - Lane deskew during normal operation (that is, after packet data transfer)
- The IP does not support the following Frame Synchronization schemes:
  - Digital Signal Processor (DSP) controller
  - CMOS Image Sensor (CIS) controller without shared clock
- The IP does not support extraction of pad codes with only one to three MPCS Rx clock cycles of active payload in between:
  - Number of bytes per MPCS Rx clock cycle is dependent on the *Baud Grade* selected. Refer to the *MPCS Output Data Width* description in the Byte-to-Pixel Converter section.
- When UVSI is enabled, the datapath configuration is fixed at generation time and the design synthesizes only the selected configuration.
  - The datapath does not support dynamic reconfiguration of the following parameters during operation:
    - LANE_WIDTH
    - PIXEL_BIT
    - BAUD_GRADE
    - FIFO_DELAY
  - UVSI testbench support is not available.
- PHY control-code detection is based on decoded data values. In rare cases, payload data may contain byte patterns that match configured PHY control codes and be interpreted as control symbols, resulting in a CRC error and possible packet corruption. Reception automatically re-synchronizes at the next valid Start code, typically limiting the impact to a single packet.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# References

- SLVS-EC Receiver Release Notes (FPGA-RN-02121)
- CertusPro-NX SERDES/PCS User Guide (FPGA-TN-02245)
- Lattice Memory Mapped Interface and Lattice Interrupt Interface User Guide (FPGA-UG-02039)
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- MPCS Module User Guide (FPGA-IPUG-02118)
- SLVS-EC Receiver IP Core web page
- CertusPro-NX web page
- Lattice Radiant Software web page
- Lattice Solutions IP Cores web page
- Lattice Solutions Reference Designs web page
- Lattice Insights web page for Lattice Semiconductor training courses and learning plans

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# 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.

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

# 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 2.1, IP v1.2.4, October 2026**

| Section | Change Summary |
|---|---|
| All | • Added the IP version information on the cover page.<br>• Added a note on the IP version in the *Quick Facts* and *Revision History* sections.<br>• Made editorial fixes. |
| Abbreviations in This Document | Updated this section. |
| Introduction | • Updated Table 1.1. Summary of the SLVS-EC Receiver IP and Table 1.3. Ordering Part Number.<br>• Added UVSI support to the Features section.<br>• Added the IP Support Summary and Attribute Names sections.<br>• Updated the license type from single machine annual and multi-site perpetual to single seat annual and single seat perpetual respectively in Table 1.3. Ordering Part Number.<br>• Removed the *IP Validation Summary* section. |
| Functional Description | • Updated Figure 2.1. Lattice SLVS-EC Receiver Block Diagram and Figure 2.2. Clock Domain Crossing Diagram.<br>• Updated the output clock description in the Clocking Overview section.<br>• Added the *Device Transmitter* interface to Table 2.2. User Interfaces and Supported Protocols.<br>• Added the Pixel Group column to Table 2.3. Byte-to-Pixel Converter Data Groups.<br>• Replaced *Data Group* with *Pixel Group* in the *FIFO Delay* formula in the Byte-to-Pixel Converter section.<br>• Added the SLVS-EC Receiver UVSI section. |
| IP Parameter Description | Added the *Enable Unified Video Streaming Transmitter Interface* attribute to Table 3.1. General Attributes. |
| Signal Description | • Updated the descriptions for *d_p_i[N-1:0]* and *d_n_i[N-1:0]* in Table 4.2. Primary I/O Ports.<br>• Updated the description for *pixel_clk_i* in Table 4.3. Native Interface Ports.<br>• Added Table 4.4. Byte-to-Pixel Data Width by Baud Grade and Number of Rx Lanes.<br>• Updated the description for *hdr_type_o[2:0]* in Table 4.5. Packet Header Information Ports.<br>• Added the Unified Video Streaming Interface section. |
| Register Description | In Table 5.2. Summary of SLVS-EC Receiver Registers:<br>• Updated the bit widths for the *INITIAL_LENGTH* register.<br>• Updated the default values for the *LINE_LENGTH* and *PIXEL_BIT* registers. |
| Designing with the IP | • Added a note on screenshots in this section.<br>• Updated the Timing and Physical Design Constraints section. |
| Appendix B. Limitation | Added UVSI and PHY control-code detection limitations. |
| References | Added the *SLVS-EC Receiver Release Notes (FPGA-RN-02121)*, *SLVS-EC Receiver IP Core* web page, and *Lattice Solutions Reference Designs* web page. |

**Revision 2.0, August 2024**

| Section | Change Summary |
|---|---|
| All | • Made editorial fixes.<br>• Updated *SerDes* to *SERDES*. |
| Abbreviations in This Document | • Replaced *acronyms* with *abbreviations* in this section.<br>• Added the following abbreviations:<br>  • *Complementary Metal-Oxide Semiconductor (CMOS)*<br>  • *Device Under Test (DUT)*<br>  • *Intellectual Property (IP)*<br>  • *Least Significant Bit (LSB)*<br>  • *Multi-Protocol PCS (MPCS)* |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

| Section | Change Summary |
|---|---|
| | • *Receiver (Rx)*<br>• *Serializer Deserializer (SERDES)*<br>• *Transmitter (Tx)* |
| Introduction | • Moved introductory paragraph in the 1. Introduction section to the 1.1. Overview of the IP section and updated the heading numbers of remaining sections accordingly.<br>• Updated Table 1.1. Summary of the SLVS-EC Receiver IP.<br>• Removed *Not Supported* features in the 1.3. Features section.<br>• Moved the content from previous *4.1. Licensing the IP* and *4.4. Hardware Validation* sections to the 1.4. Licensing and Ordering Information section and updated its content.<br>• Moved the content from previous *5. Ordering Part Number* section to the 1.4.1. Ordering Part Number section and updated its content.<br>• Added the 1.5. IP Validation Summary and 1.6. Minimum Device Requirements sections, and updated the heading numbers of remaining sections accordingly.<br>• Renamed the previous *1.3. Conventions* section to *1.7. Naming Conventions* and removed the *Attribute* information. |
| Functional Description | • Moved the content from previous *2.1. General Description* and *2.2. Block Diagram* sections to the 2.1. IP Architecture Overview section and updated its content.<br>• Moved the content from previous *2.2.5. Clock, Reset and Initialization* section to the 2.2. Clocking and 2.3. Reset sections and updated their contents.<br>• Added the 2.4. User Interfaces section and updated the heading numbers of remaining sections accordingly.<br>• Removed the previous *2.2.3. AXI4-Stream Device Transmitter* section.<br>• Moved the previous *3.3. Timing Descriptions* section to the 2.8. Timing Descriptions section. |
| IP Parameter Description | Moved the content from previous *3.1. Attributes Summary* section to this section and updated its content. |
| Signal Description | Moved the content from previous *3.0. Signal Description* section to this section and updated its content. |
| Register Description | Moved the content from previous *3.2. Register Description* section to this section and updated its content. |
| Designing with the IP | • Moved the content from previous *4.2. Generation and Synthesis* section to the 6.1. Generating and Instantiating the IP section and updated its contents including all the figures.<br>• Added the following subsections and updated the heading numbers of remaining sections accordingly:<br>  • *6.2. Design Implementation*<br>  • *6.3. Timing and Physical Design Constraints*<br>  • *6.4. Specifying the Strategy*<br>• Updated the contents including all the figures in the 6.5. Running Functional Simulation section. |
| Debugging | Added this section. |
| Appendix A. Resource Utilization | Updated this section. |
| Appendix B. Limitation | Moved the previous *1.2.2. Not Supported* section to this section and updated its content. |
| References | • Added the following references:<br>  • *Lattice Radiant Software web page*<br>  • *Lattice Solutions IP Cores web page*<br>• Removed the following references:<br>  • *Certus-NX web page*<br>  • *CrossLink-NX web page* |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

**Revision 1.9, January 2024**

| Section | Change Summary |
|---|---|
| All | • Renamed the document from *SLVS-EC Receiver IP Core - Lattice Radiant Software* to *SLVS-EC Receiver IP*.<br>• Made editorial fixes. |
| Disclaimers | Updated boilerplate. |
| Introduction | • Moved previous key features in Features section into the Supported section.<br>• Added Not Supported section. |
| Functional Description | Added the following paragraph in the Rx Protocol Management FSM section:<br>*Current IP only supports lane deskew during initialization/training phase. In cases when lane skew occurs during packet data transfer, you can perform the following procedure:*<br>*1. In the SLVS-EC Rx soft IP, add a custom deskew logic after the MPCS IP instance that will search for the START code from packet data of each lane.*<br>*2. Add single clock FIFO that will store these packets after START is detected.*<br>*3. Read out from FIFO after START is detected in all lanes.* |
| Signal Description | Updated the Selectable and Default values in Table 3.2. Attributes Table of the following attributes:<br>• *Native Interface*<br>• *AXI4-Stream Transmitter Interface*<br>• *Manual Adjust*<br>• *Pixel Count*<br>• *FIFO Depth*<br>• *FIFO Delay*<br>• *Header Analysis*<br>• *Miscellaneous Signals* |
| Ordering Part Number | Updated this section. |
| References | • Updated the names of existing references.<br>• Added references to *Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)* and *Lattice Insights* web page. |

**Revision 1.8, February 2023**

| Section | Change Summary |
|---|---|
| Functional Description | • Updated Figure 2.7. Byte-to-Pixel Converter Implementation to revise 'MPCS clock domain' with 'MPCS Rx clock domain'.<br>• Updated Figure 2.2. Clock Domain Crossing Diagram to replace 'mpcs_clk_i' with 'mpcs_clkin_i'.<br>• Replaced ‘MPCS Clock’ with ‘MPCS Rx Clock’.<br>• Replaced 'AXI4-Stream Master' with 'AXI4-Stream Transmitter'.<br>• Updated Table 2.1. Pre-Configured MPCS PLL Setting for SLVS-EC Rx to add the new column 'MPCS Rx Clock (MHz)'. |
| Signal Description | • Updated Table 3.2. Attributes Table to replace 'MPCS Clock Frequency' to 'Reference Clock Frequency'.<br>• Replaced 'AXI4-Stream Master' with 'AXI4-Stream Transmitter'. |
| IP Generation, Simulation, and Validation | Updated Figure 4.2. Configure User Interface of SLVS-EC Rx IP. |

**Revision 1.7, September 2022**

| Section | Change Summary |
|---|---|
| Functional Description | • Updated the calculating formulas and added sample computations to the SLVS-EC Rx Core section. |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

| Section | Change Summary |
|---|---|
| | • Changed the description to *The MPCS clock used in* Table 3.2 *is the default 125 MHz* in the Clock Domain section.<br>• Changed the description to *The default MPCS clock displayed in the IP user interface is 125 MHz* in the Adjusting Reference Clock for Different Baud Rate section. |
| Signal Description | • Updated mpcs_rx_clk_o signal description removing 125 MHz in Table 3.1. SLVS-EC Rx IP Signal Description.<br>• Updated the *Data Rate* group of data in Table 3.2. Attributes Table and Table 3.3. Attributes Descriptions. |
| Ordering Part Number | Newly added the SLVS-EC-RX-CPNX-US part. |

**Revision 1.6, May 2022**

| Section | Change Summary |
|---|---|
| Introduction | Revised feature to *Supports PHY layer functions using the embedded PMA/PCS in CertusPro-NX FPGA device*. |

**Revision 1.5, January 2022**

| Section | Change Summary |
|---|---|
| Functional Description | • Updated MPCS Clock information in Clock Domain.<br>• Added Adjusting Reference Clock for Different Baud Rate section. |
| References | Added reference link to CertusPro-NX SerDes/PCS Usage Guide. |

**Revision 1.4, November 2021**

| Section | Change Summary |
|---|---|
| Introduction | Updated Table 1.1 to change IP version to v1.1.x and Radiant version to 3.0 or later. |
| Functional Description | Updated FIFO equation in Byte-to-Pixel Converter. |
| Appendix A. Resource Utilization | Updated Register and LUT values in Table A.1. |

**Revision 1.3, June 2021**

| Section | Change Summary |
|---|---|
| All | • Minor adjustments in formatting.<br>• Added CertusPro-NX across the document, |
| Introduction | Updated Table 1.1. |
| Signal Description | • Updated Table 3.1 to remove LINTR row.<br>• Updated Table 3.4 to remove Interrupt Status and Status Register rows and two table notes. |
| Ordering Part Number | Added this section. |
| Appendix A. Resource Utilization | Updated Radiant version. |

**Revision 1.2, April 2021**

| Section | Change Summary |
|---|---|
| All | Minor adjustments in formatting. |
| Acronyms in This Document | Added this section. |
| Signal Description | • Updated Table 3.1, Table 3.2, and Table 3.3 to remove Frame Synchronization Signals sections.<br>• Updated Table 3.4 to add table note 4 and 5. |
| IP Generation, Simulation, and Validation | Updated Figure 4.2. |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.

| Section | Change Summary |
|---|---|
| References | Updated section content. |

**Revision 1.1, March 2021**

| Section | Change Summary |
|---|---|
| Introduction | Updated Table 1.1 to change IP v1.0.0 to *v1.0.x*. |
| IP Generation, Simulation, and Validation | Updated Running Functional Simulation section content. |

**Revision 1.0, December 2020**

| Section | Change Summary |
|---|---|
| All | Initial release. |

© 2026 Lattice Semiconductor Corp. All Lattice trademarks, registered trademarks, patents, and disclaimers are as listed at www.latticesemi.com/legal.
All other brand or product names are trademarks or registered trademarks of their respective holders. The specifications and information herein are subject to change without notice.




www.latticesemi.com