SubLVDS Image Sensor Receiver IP

IP Version: 1.7.1

User Guide

FPGA-IPUG-02093-2.2

December 2025

Disclaimers

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

Inclusive Language

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

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
APB	Advanced Peripheral Bus
AXI4-Stream	Advanced eXtensible Interface 4 Stream
DUT	Device Under Test
EAV	End of Active Video
HDL	Hardware Description Language
IMX	Sony Image Sensor Family
IP	Intellectual Property
LSE	Lattice Synthesis Engine
LVDS	Low-Voltage Differential Signaling
RX	Receiver
SAV	Start of Active Video
SubLVDS	A reduced-voltage form of LVDS signaling
TX	Transmitter
XHS	Horizontal Sync
XVS	Vertical Sync

1. Introduction

1.1. Overview of the IP

The Lattice™ Semiconductor SubLVDS Image Sensor Receiver IP Core converts double data rate interface to pixel clock domain. The SubLVDS interface is primarily used in image sensors. The interface has one clock pair and more than one data pairs. The number of data pairs varies, depending on bandwidth requirement. SubLVDS is a source synchronous interface, the clock pair is running at the same rate as the data. This is not a 7:1 interface. SubLVDS has the clock center-aligned with the data.

Table 1.1. Comparison Between the LVDS Interface and SubLVDS Interface

Feature	LVDS Interface	SubLVDS Interface
Common mode voltage	1.25 V	0.9 V
Power supply	2.5 V	1.8 V
Differential swing	±175 mV	±150 mV

1.2. Quick Facts

Table 1.2. SubLVDS Image Sensor Receiver IP Core Quick Facts

IP Requirements	Supported Devices	Lattice Avant™, MachXO5™-NX, CrossLink™-NX, CertusPro™-NX, Certus™-NX, Certus-N2
	IP Changes[1]	For a list of changes to the IP, refer to the SubLVDS Image Sensor Receiver IP Release Notes (FPGA-RN-02051).
Resource Utilization	Supported User Interface	Native Interface, APB Interface, and AXI4-Stream Transmit Interface. See the Signal Description section.
	Resources	See the Resource Utilization section
Design Tool Support	Lattice Implementation	IP Core v1.7.1 – Lattice Radiant™ software 2025.2
	Synthesis	Lattice Synthesis Engine (LSE)
		Synopsys® Synplify Pro® for Lattice
	Simulation	For the 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 1.3. SubLVDS Image Sensor Receiver IP Support Readiness

The device families in this table have the following features in common:

- Number of RX lanes: 4,8,10,12,16
- RX gear: 8, 16
- RX line rate: 160, 500

Device Family	Dropped Line Mode	Dropped Line Count	Dropped Pixel Mode	Dropped Pixel Count	Word Count Mode	Word Count	APB Interface	AXI4-Stream TX Interface	Data Type	Radiant Timing Model	Hardware Validation
Avant	Dynamic	—	Dynamic	—	Dynamic	—	Off, On	Off, On	RAW10 RAW12	Preliminary	No
	Static	0, 2, 7	Static	0, 2, 7	Static	0, 2, 7					
CertusPro-NX	Dynamic	—	Dynamic	—	Dynamic	—	Off, On	Off, On	RAW10 RAW12	Preliminary	No
	Static	0, 2, 7	Static	0, 2, 7	Static	0, 2, 7					

Device Family	Dropped Line Mode	Dropped Line Count	Dropped Pixel Mode	Dropped Pixel Count	Word Count Mode	Word Count	APB Interface	AXI4-Stream TX Interface	Data Type	Radiant Timing Model	Hardware Validation
CrossLink-NX	Dynamic	—	Dynamic	—	Dynamic	—	Off, On	Off, On	RAW10 RAW12	Preliminary	No
	Static	0, 2, 7	Static	0, 2, 7	Static	0, 2, 7					

1.4. Features

The key features of the SubLVDS Image Sensor Receiver IP Core include:

- Supports 4, 6, 8, 10, 12, 14, or 16 data lanes from an image sensor.
- Supports 10-bit (RAW10) or 12-bit (RAW12) pixel widths.
- Supports gearing of 8 and 16. The gearing 16 option is only for 4-lane configuration.
- Supports APB interface for register access and AXI4-Stream Transmit interface.
- Can generate XVS and XHS for image sensors operating in Passive mode.

1.5. Licensing and Ordering Information

The SubLVDS Image Sensor Receiver IP is available with the Lattice Radiant Subscription software. To purchase the Lattice Radiant Subscription license, contact Lattice Sales or go to the Lattice Online Store.

1.6. Minimum Device Requirements

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

1.7. Naming Conventions

1.7.1. Nomenclature

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

1.7.2. Signal Names

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

2. Functional Description

2.1. IP Architecture Overview

The SubLVDS Image Sensor Receiver IP Core converts double data rate interface into pixel clock domain. The input interface of the design consists of a data bus and a clock in SubLVDS interface format. The output interface consists of a 10-bit or 12-bit multi-pixel data, frame valid, line valid, data valid, and a pixel clock with a gearing of 1:8 or 1:16.

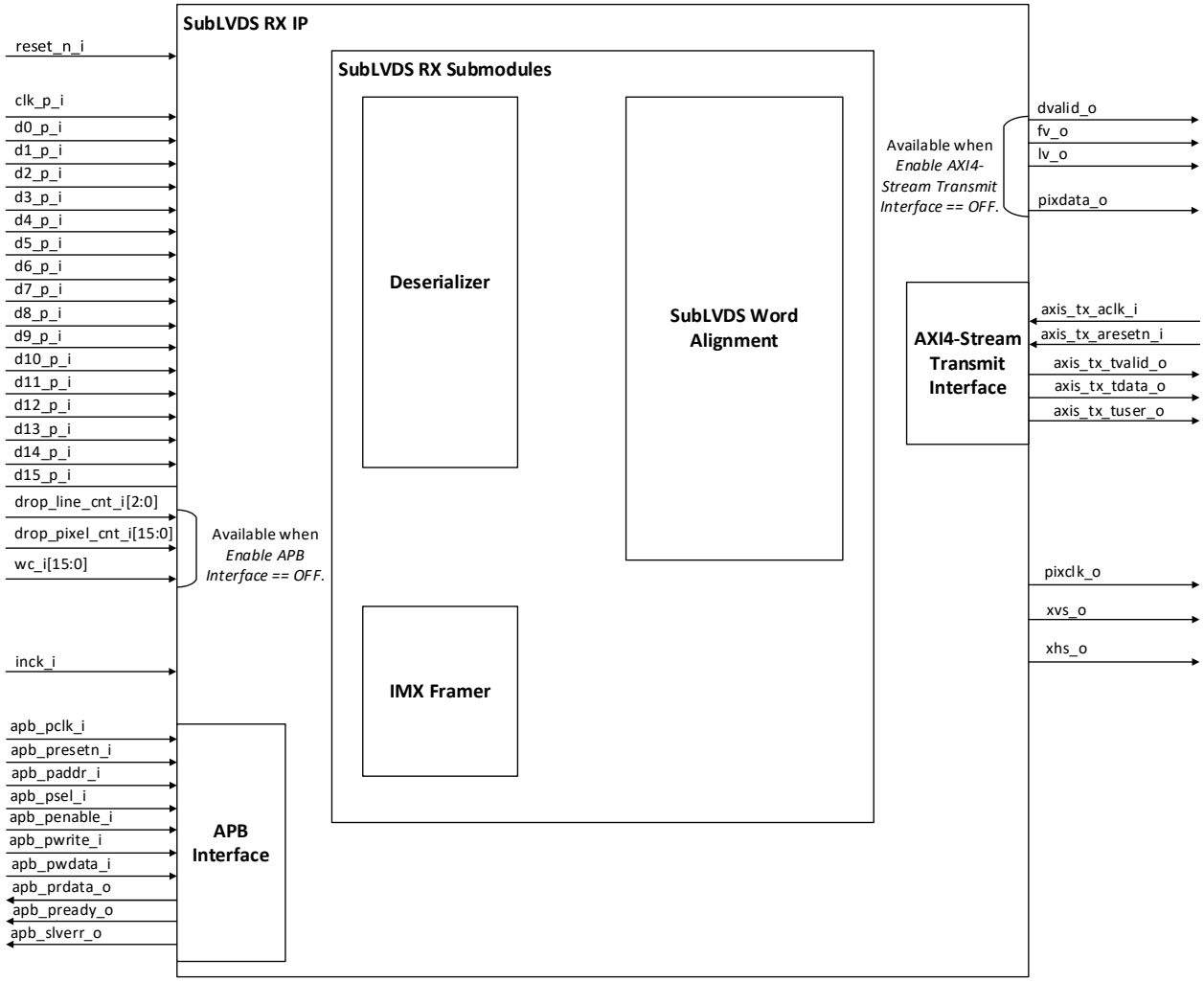

Figure 2.1. SubLVDS Image Sensor Receiver IP Core Top Level Block Diagram

2.2. Clocking

The RX clock input, clk_p_i, is from an external source (image sensor) and must be connected to a dedicated SubLVDS edge clock pin. The Deserializer block generates a pixel clock, pixclk_o, with a gearing of 1:8 or 1:16 for the pixel data.

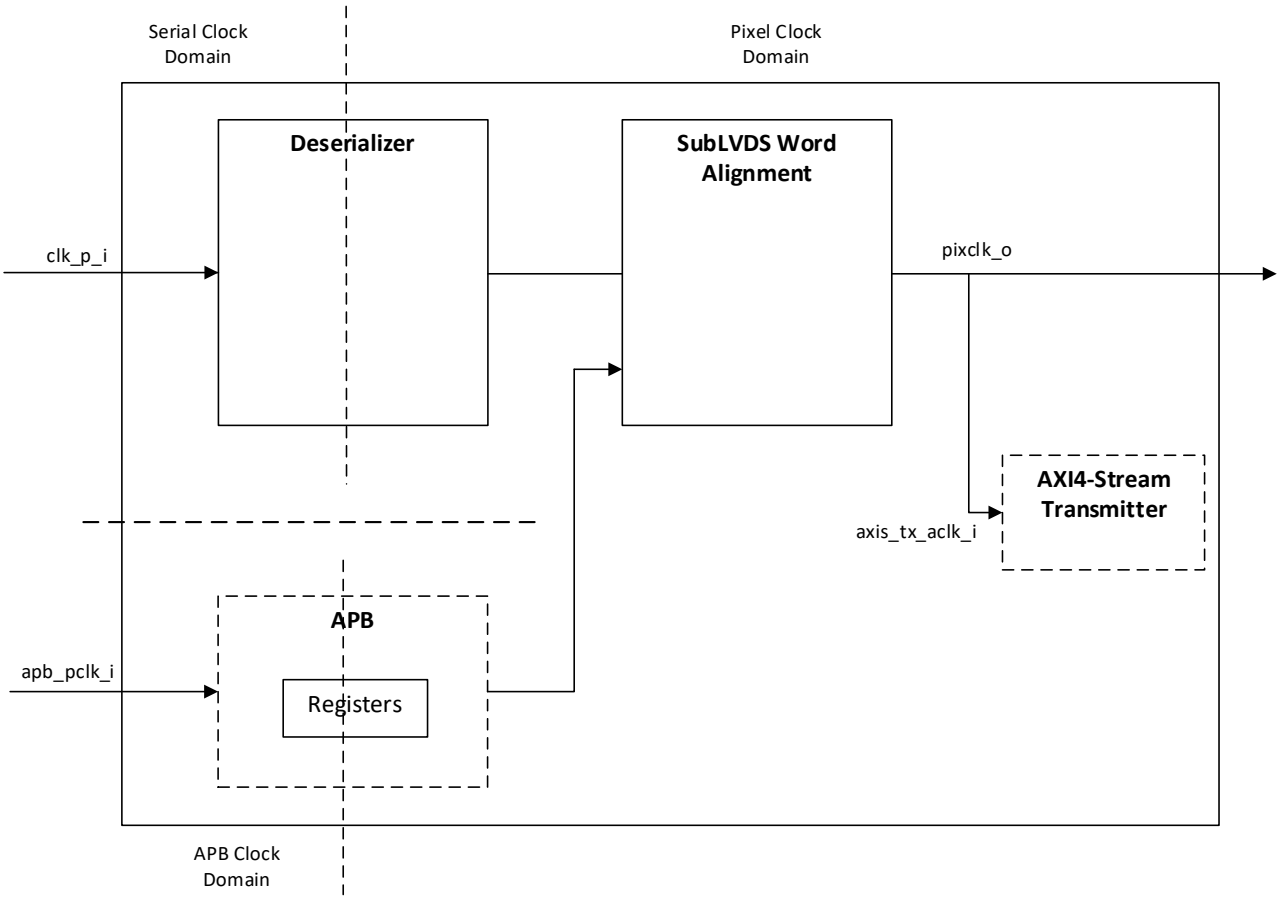

Figure 2.2. Clock Domain Crossing Block Diagram

Table 2.1. Clock Domain Crossing

Clock Domain Crossing	Handling Approach
SubLVDS Serial Clock to Pixel Clock	1:8/1:16 gearbox DDR Hard IP

The general formula for computing the required clocks of the system:

RX Line Rate (total) = Total pixels in a frame (active + blanking) ✕ Frame rate ✕ Bits per pixel

RX Line Rate (per lane) = RX Line Rate (total)/Number of RX lanes

RX input clock = RX Line Rate (per lane)/2

Pixel clock = RX input clock/Gearing

Note: Gearing = 4 for 1:8 gearing; Gearing = 8 for 1:16 gearing.

2.3. Reset

Active low reset is used in the design with synchronous release. Resets for each clock domain are synchronized to the respective clock domains.

The system reset, reset_n_i, is synchronized to the pixel clock domain and serves as a reset source for the SubLVDS Word Alignment module.

No special reset sequence is required in this IP.

2.4. User Interfaces

Table 2.2. User Interfaces and Supported Protocols

User Interface	Supported Protocols	Description
Device Transmitter	AXI4-Stream	The AXI4-Stream Transmit interface provides the transmission of pixel data. The AXI4-Stream data valid acts as an enable signal and the data is driven on the AXI transmit data bus.
Control	APB	Configures Pixel Drop Count, Line Drop Count, and Word Count dynamically. Refer to the Register Description section for more information about the registers.

2.4.1. AXI4-Stream Transmit Interface

2.4.1.1. Default Normal Transmission

The following figure shows the timing diagram of default normal transmission when using the AXI4-Stream interface:

- T0: *axis_tx_tuser_o*[1:0] and *axis_tx_tvalid_o* are asserted to indicate the start of valid data transmission.
- T1: *axis_tx_tuser_o*[0] is de-asserted to indicate the end of valid data transmission.

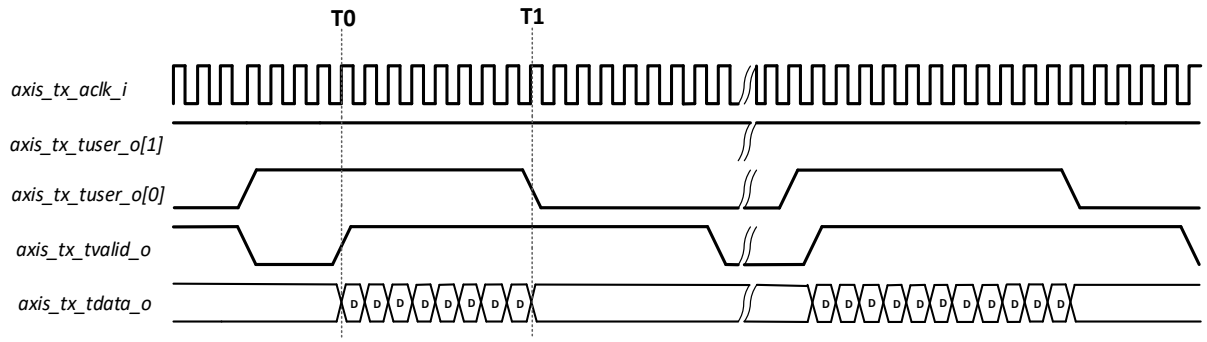

Figure 2.3. Timing Diagram of Default Normal Transmission

2.4.1.2. Custom Transmission

AXI4-Stream Transmission allows only data width in multiples of 8 (refer to *AMBA 4 AXI4-Stream Protocol version 1.0* for the operation). In this transmission, data padding is used when the Number of Rx Lanes == 6, 10, and 14 for the Data Type == RAW10.

The following figure shows the timing diagram of custom transmission when using the AXI4-Stream interface.

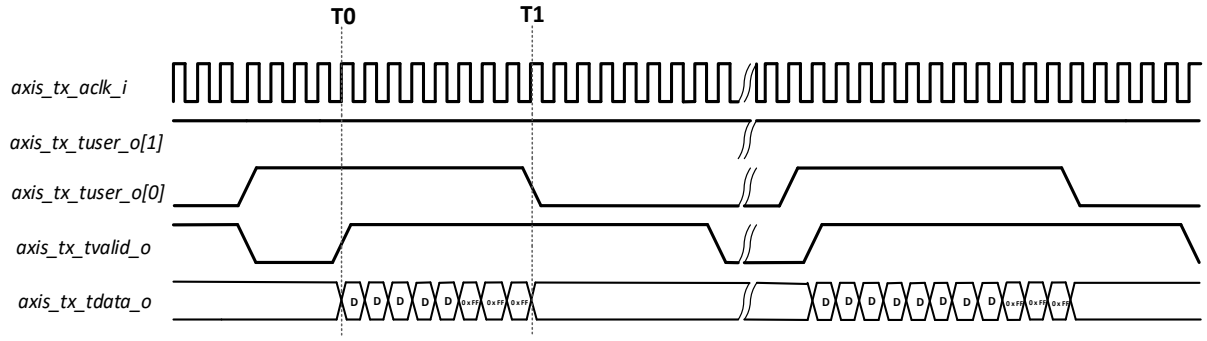

Figure 2.4. Timing Diagram of Custom Transmission

If AXI4-Stream Transmitter is not enabled, the following internal signals turn to top-level output signals.

- pixdata_o
- fv_o
- lv_o
- dvalid_o

2.5. SubLVDS Image Sensor Receiver IP Core Submodules

The SubLVDS RX IP consists of the following modules:

- Deserializer
- SubLVDS Word Alignment
- IMX Framer
- APB Interface (optional module)
- AXI4-Stream Transmitter Interface (optional module)

The Deserializer block converts each double data rate lane (d*_p_i signals) to a single data rate 8-bit or 16-bit at a slower operating speed within a system.

The Word Alignment module receives the 8-bit (1:8 gearing) or 16-bit (1:16 gearing) deserialized data (deser_q_o signal) and converts the data to 10-bit or 12-bit pixel data according to the set configuration of data type (RAW10 or RAW12). The output of the module is a multi-pixel bus (pixdata_o), pixel clock (pixclk_o), dvalid_o, fv_o, and lv_o control signals.

The IMX Framer module is used for Image Sensors that operate in the Passive mode only.

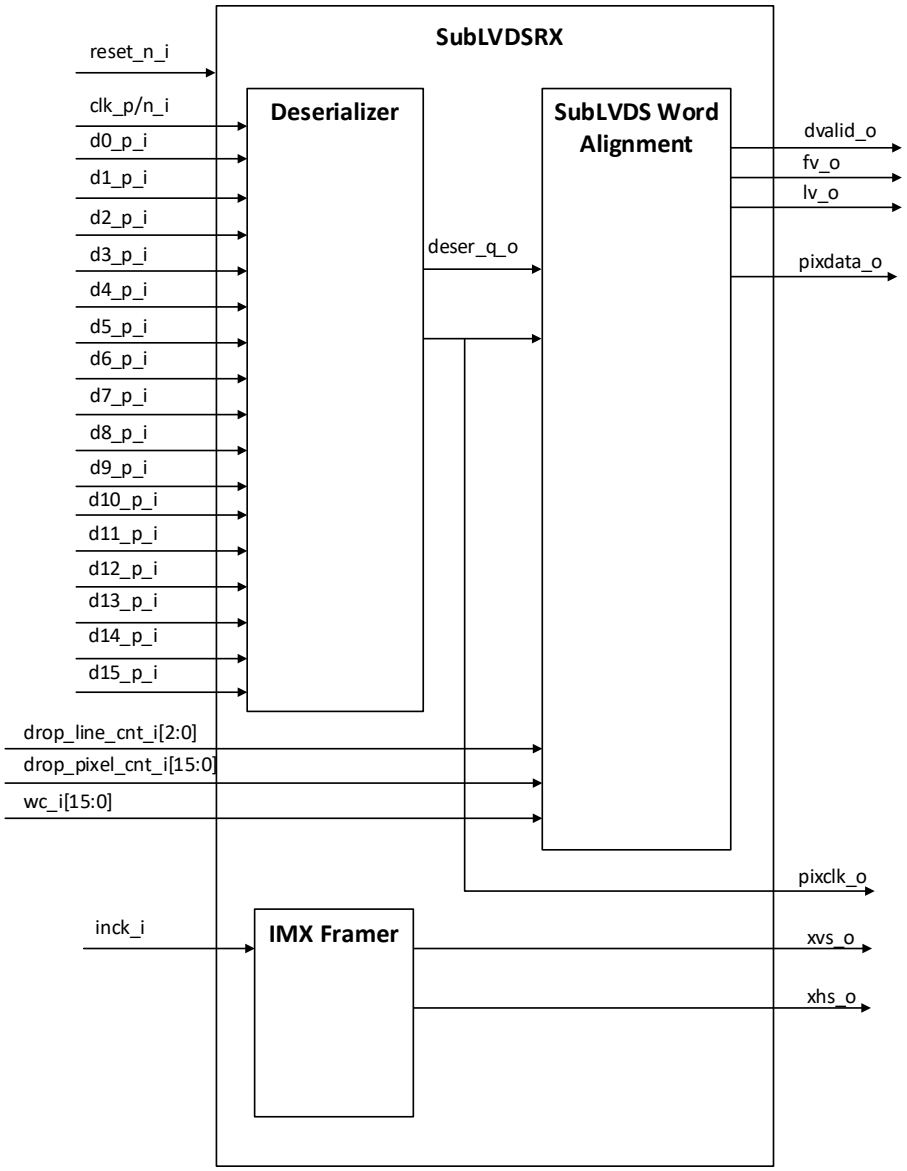

Figure 2.5. SubLVDS Image Sensor Receiver IP Core Detailed Block Diagram (APB and AXI4-Stream Transmitter Interfaces Not Enabled)

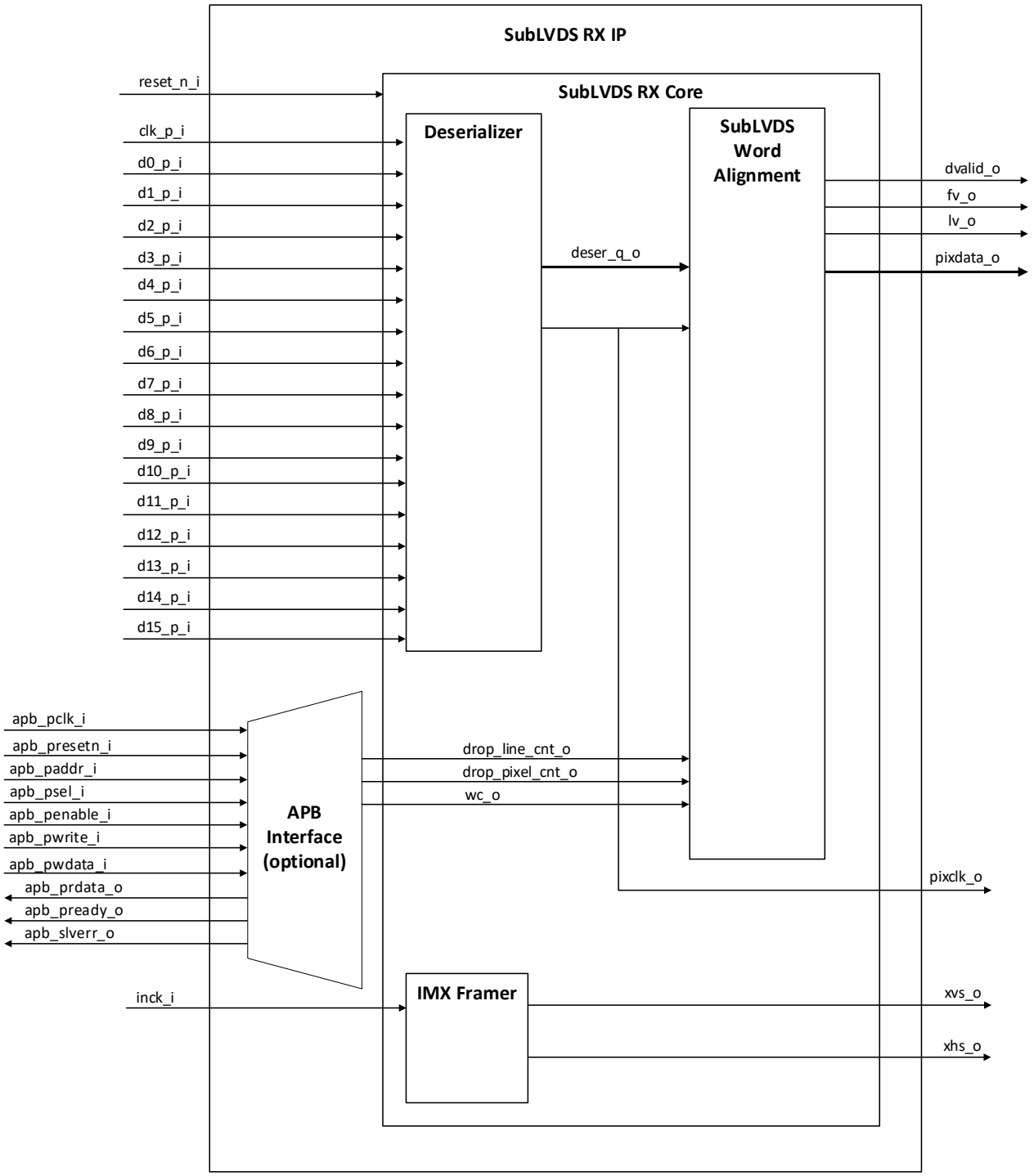

Figure 2.6. SubLVDS Image Sensor Receiver IP Core Detailed Block Diagram (APB Interface Enabled)

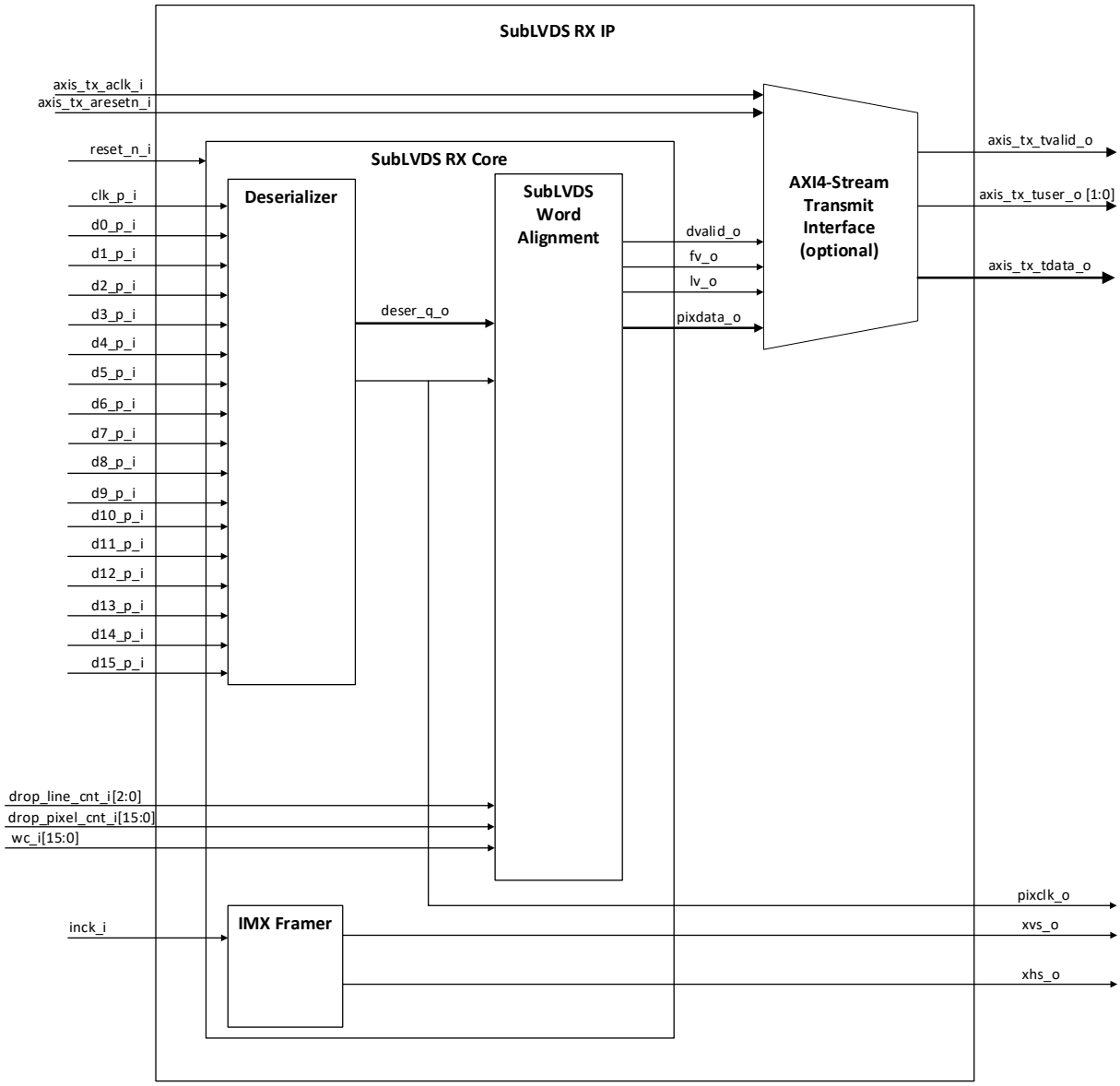

Figure 2.7. SubLVDS Image Sensor Receiver IP Core Detailed Block Diagram (AXI4-Stream Transmitter Interface Enabled)

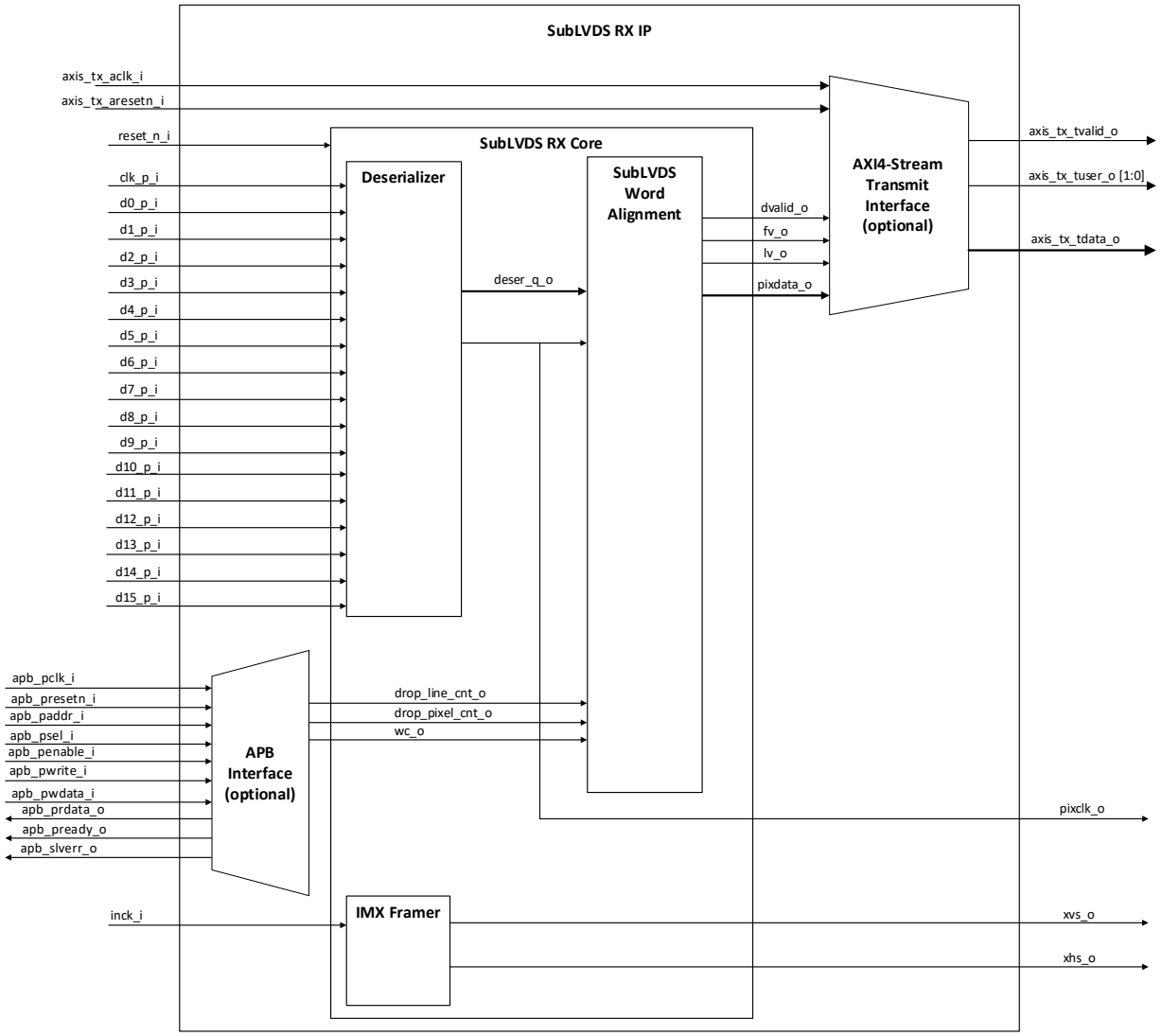

Figure 2.8. SubLVDS Image Sensor Receiver IP Core Detailed Block Diagram (APB and AXI4-Stream Transmitter Interfaces Enabled)

2.6. Timing Specifications

The example shown in the following figure has an effective readout pixel of 4096 by 2160.

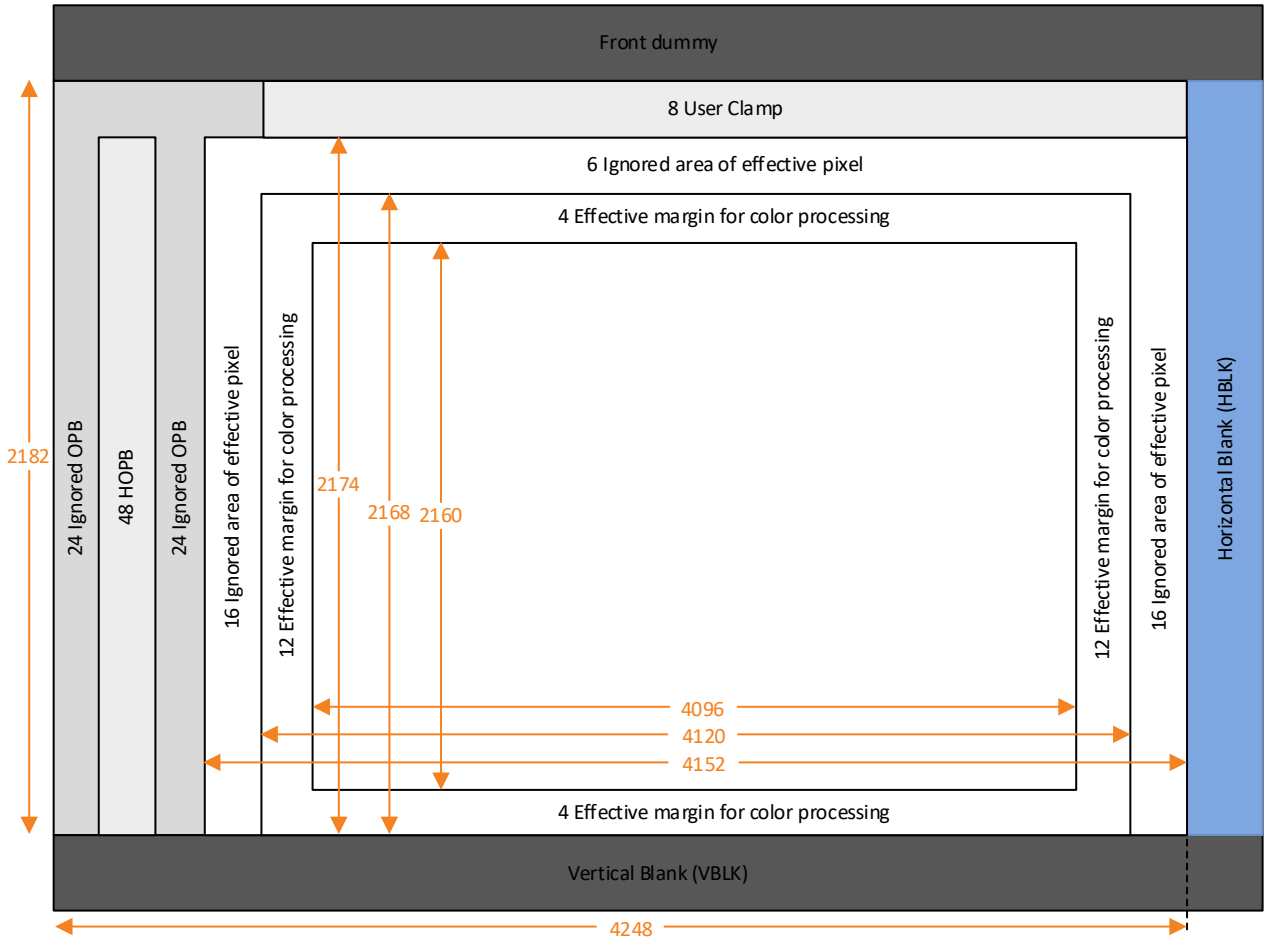

Figure 2.9. Example of SubLVDS Frame Diagram of a Sensor

The following timing diagram of the SubLVDS Image Sensor Receiver IP Core input interface is for the frame diagram example shown in the figure above. The timing diagram shows the sync signal and data output timing during 10-bit length serial received from the image sensor.

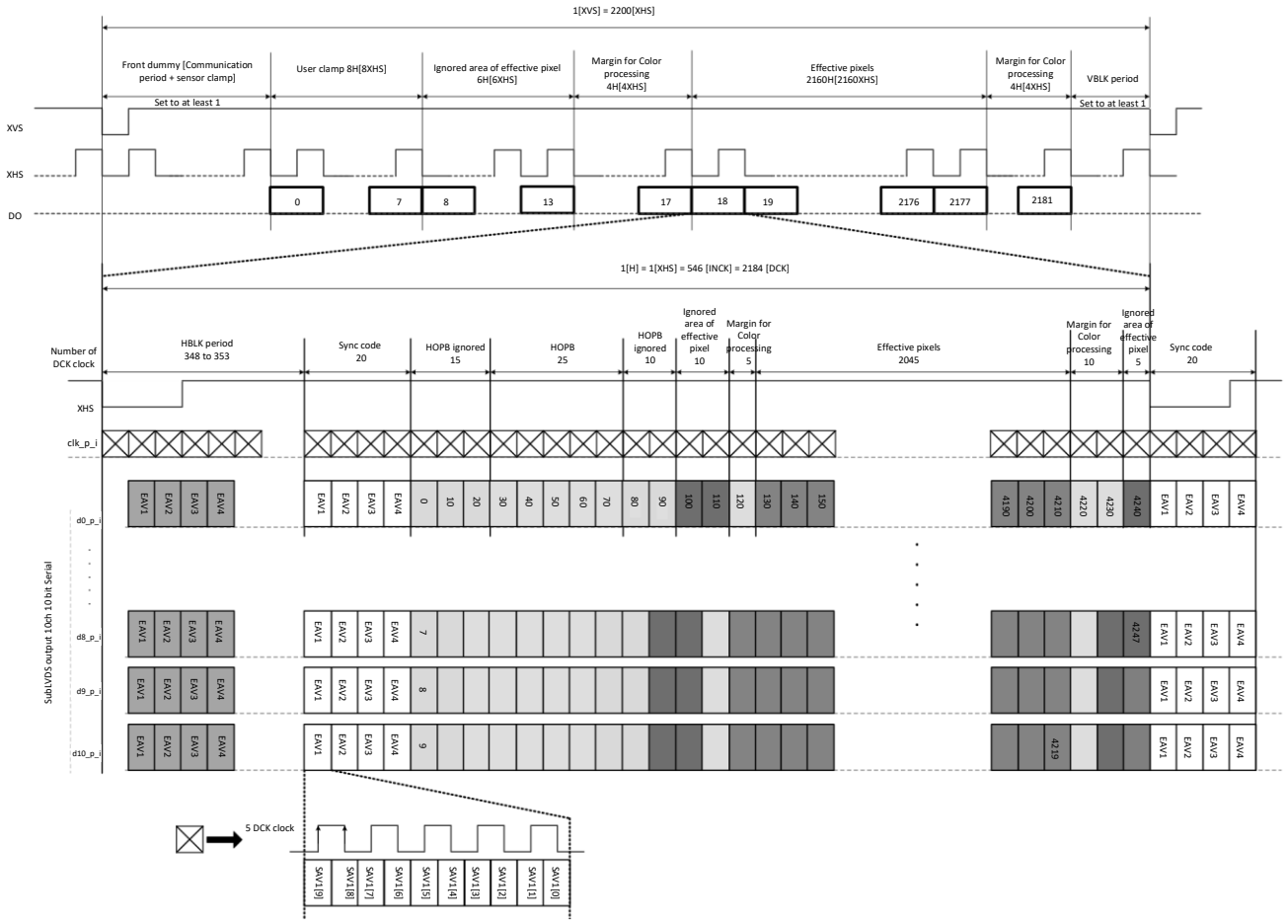

Figure 2.10. SubLVDS Image Sensor Receiver IP Core Input Bus Waveform

The horizontal and vertical timing of the received data are denoted by the length of XHS and XVS sync signals respectively. The sync codes (SAV and EAV) are added before and after the payload data, such as effective pixels and HOPB. The following tables list the sync code details for both 10-bit and 12-bit variants. Note that the frame and timing diagrams are dependent on the sensors and configurations. Refer to the specific sensor datasheet for more information.

Table 2.3. Sync Codes

LVDS Output Bit No.		Sync Code			
12-bit Output	10-bit Output	1st Word	2nd Word	3rd Word	4th Word
11	9	1	0	0	1
10	8	1	0	0	0
9	7	1	0	0	V[1]
8	6	1	0	0	H[2]
7	5	1	0	0	P3[3]
6	4	1	0	0	P2[3]
5	3	1	0	0	P1[3]
4	2	1	0	0	P0[3]
3	1	1	0	0	0
2	0	1	0	0	0
1	—	1	0	0	0
0	—	1	0	0	0

Table 2.4. Protection Bits

V and H Values		Protection Bits			
V[1]	H[2]	P3[3]	P2[3]	P1[3]	P0[3]
0	0	0	0	0	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	0

Notes:

1. V value: 1: Blanking line; 0: Except blanking line.
2. H → 1: End sync code; 0: Start sync code.
3. P0 – P3 → Protection bits that are dependent on the V and H values of the current packet.

The following tables show the full values of the words by stitching the bits from Table 2.3 and Table 2.4 for 12-bit and 10-bit serial output respectively. Based on the timing diagram in Figure 2.10, the word mapping is as follows:

- 1st Word of SAV corresponds to SAV1
- 2nd Word of SAV corresponds to SAV2
- 3rd Word of SAV corresponds to SAV3
- 4th Word of SAV corresponds to SAV4

This word mapping works similarly for the EAVs.

Table 2.5. Sync Code Details (Hexadecimal Notation) for 12-bit Serial Output

Sync Signal	Sync Code	1st Word	2nd Word	3rd Word	4th Word
Blanking Line	Start Sync Code (SAV)	12'hFFF	12'h000	12'h000	12'hAB0
	End Sync Code (EAV)				12'hB60
Except Blanking Line	Start Sync Code (SAV)				12'h800
	End Sync Code (EAV)				12'h9D0

Table 2.6. Sync Code Details (Hexadecimal Notation) for 10-bit Serial Output

Sync Signal	Sync Code	1st Word	2nd Word	3rd Word	4th Word
Blanking Line	Start Sync Code (SAV)	10'h3FF	10'h000	10'h000	10'h2AC
	End Sync Code (EAV)				10'h2D8
Except Blanking Line	Start Sync Code (SAV)				10'h200
	End Sync Code (EAV)				10'h274

2.7. Sample Configurations

The following waveforms show the output behaviors for *Number of Rx Lanes* == 4, Data Type == RAW10, and the number of pixels sent by the sensor is 40 pixels, with different *Word Count* and *Dropped Pixel Count*. Note that the IP interprets the metadata (such as Clamp, OPB) the same as the effective pixel data. If the metadata is not needed, you can configure Word Count and Dropped Pixel Count accordingly. Otherwise, you need to design custom logics to consume and process the metadata.

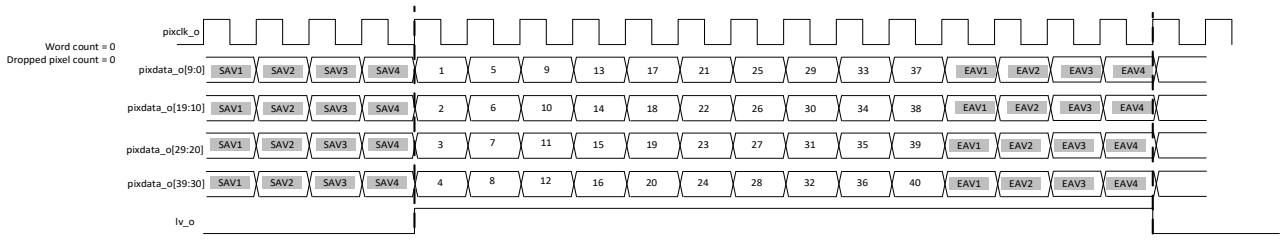

Figure 2.11. SubLVDS Image Sensor Receiver Output Concept Waveform when Word Count == 0 and Dropped Pixel Count == 0

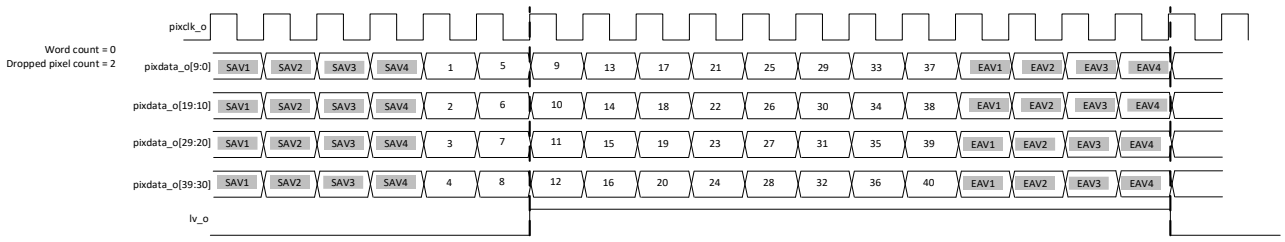

Figure 2.12. SubLVDS Image Sensor Receiver Output Concept Waveform when Word Count == 0 and Dropped Pixel Count == 2

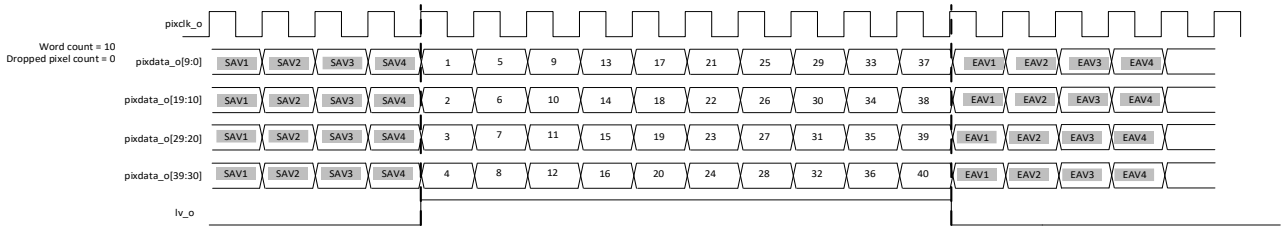

Figure 2.13. SubLVDS Image Sensor Receiver Output Concept Waveform when Word Count == 10 and Dropped Pixel Count == 0

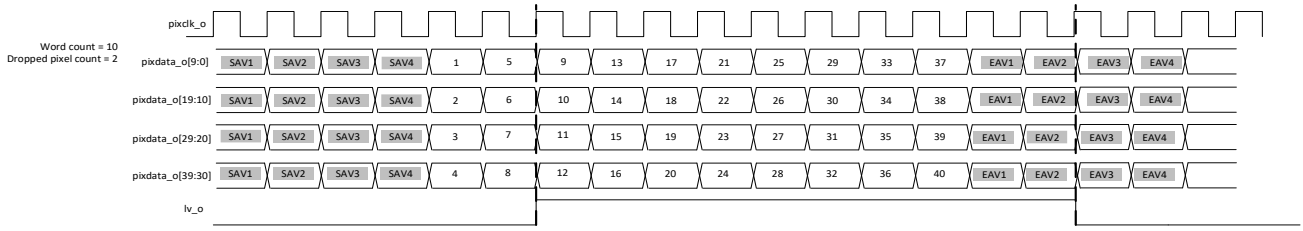

Figure 2.14. SubLVDS Image Sensor Receiver Output Concept Waveform when Word Count == 10 and Dropped Pixel Count == 2

3. IP Parameter Description

The configurable attributes of the SubLVDS Image Sensor Receiver IP are shown in the following tables. 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
Receiver		
Number of RX Lanes	**4**, 6, 8, 10, 12, 14, 16	Number of subLVDS I/O lanes to be generated by IP.
RX Gear	**8**, 16	Specifies the RX Gearing. Only the 4-lanes configuration has the option to choose between *RX Gear == 8* or *RX Gear == 16*. For other configurations, *RX Gear* is fixed to *8*.
Clock		
RX Line Rate	**160** – 1250	Target RX Line Rate per lane.
	160 – 1500	1500 Mbps maximum line rate is supported in CertusPro-NX flip-chip packages only. For non-flip-chip packages, the maximum line rate is 1250 Mbps.
SubLVDS Clock Frequency (MHz)	Calculated	SubLVDS clock. Automatically computed based on the target *RX Line Rate*.
Pixel Clock Frequency (MHz)	Calculated	Pixel clock. Automatically computed based on target *RX Line Rate* and *RX Gear*.
Data		
Dropped Line Mode	Static, **Dynamic**	Defined Dropped Line mode. *Static* mode sets a pre-determined value of the number of dropped lines during compile time. *Dynamic* mode defines the values via the IP port which can be dynamically configured.
Dropped Line Count	**0** – 7	Available when *Dropped Line Mode == Static*. Determines the number of lines to be dropped at the start of the frame.
Dropped Pixel Mode	Static, **Dynamic**	Defined Dropped Pixel mode. *Static* mode sets a pre-determined value of the number of dropped pixels during compile time. *Dynamic* mode defines the values via the IP port which can be dynamically configured.
Dropped Pixel Count	**0** – 65535	Available when *Dropped Pixel Mode == Static*. Crops the number of pixels after SAV (the OPB and OPB ignore pixels). Refer to the Sony® IMX sensor specification for information on these pixels. The input value equals to the desired number of pixels to drop/*Number of RX Lanes*. For example, to drop 8 pixels when *Number of RX Lanes* == 4, the input to *Dropped Pixel Count* is 2.
Word Count Mode	Static, **Dynamic**, OFF	Defined Word Count mode. *Static* mode sets a pre-determined value of the number of word count during compile time. *Dynamic* mode defines the values via the IP port which can be dynamically configured. *OFF* is when logic is not used.
Word Count	**0** – 65535	Available when *Word Count Mode == Static*. Number of active video pixels per line after the dropped pixels (when *Dropped PixelCount* > 0). Reducing

Attribute	Selectable Values	Description
		this attribute effectively drops the OPB ignore bits right before the EAV. Refer to the Sony IMX sensor specification for information on these pixels. If *Word Count* == 0 and *Dropped Pixel Count* == 0, the total number of pixels coming outof the design is the total number of active pixels sent by sensor + EAV pixels. The input value equals to the desired total number of pixels/*Number of RX Lanes*. For example, if the desired total number of pixels is 40 and *Number of RX Lane* == 4, *Word Count* is 10.
Miscellaneous		
Enable APB Interface	ON, **OFF**	If this attribute is set to *ON*, the APB interface is used instead of the Native SubLVDS Rx interface to configure the Number of Dropped Lines, Number of Dropped Pixels, and Word Count dynamically.
Enable AXI4-Stream Transmit Interface	ON, **OFF**	If this attribute is set to *ON*, the AXI4-Stream Transmit interface is used instead of the Native SubLVDS Rx Interface.

3.2. Video

Table 3.2. Video

Attribute	Selectable Values	Description
Video Packet		
Data Type	**RAW10**, RAW12	Defines the video data type generated by the sensor.
IMX Framer Settings		
Image Sensor Mode	**Active**, Passive[1]	Sets the mode of the image sensor. In *Passive*[1] mode, enables the IMX framer.
V_TOTAL	2 – 4095, **10**	Available when *Image Sensor Mode = Passive*[1]. Sets the number of lines XVS is driven high.
H_TOTAL[2]	0 – 4095, **1285**	Available when *Image Sensor Mode = Passive*[1]. The number of INCK clocks XHS driven high is set by *H_TOTAL-V_H_BLANK*.
V_H_BLANK[2]	0 – 4095, **2**	Available when *Image Sensor Mode = Passive*[1]. Sets the number of INCK clocks XVS and XHS are driven low.

Notes:

1. Consistent with Lattice Semiconductor inclusive language policy, the terms *Active* and *Passive* are used in place of the original terms in the Sony Image Sensor Framer document. The technical definitions remain the same.
2. *H_TOTAL* must be greater than *V_H_BLANK*. Both attributes must also be set to a non-zero value to ensure valid data during normal operation.

4. Signal Description

This section describes the SubLVDS Image Sensor Receiver IP ports.

4.1. Native Interface

Table 4.1. Clock Ports

Port	Type	Description
reset_n_i	Input	System active low asynchronous reset.
inck_i[1]	Input	IMX Framer input clock. This clock is shared with the Sony Image Sensor.
clk_p_i	Input	Positive subLVDS input clock to subLVDS RX.
d0_p_i[2]	Input	Positive subLVDS input data lane 0 to subLVDS RX.
d1_p_i[2]	Input	Positive subLVDS input data lane 1 to subLVDS RX.
d2_p_i[2]	Input	Positive subLVDS input data lane 2 to subLVDS RX.
d3_p_i[2]	Input	Positive subLVDS input data lane 3 to subLVDS RX.
d4_p_i[2]	Input	Positive subLVDS input data lane 4 to subLVDS RX.
d5_p_i[2]	Input	Positive subLVDS input data lane 5 to subLVDS RX.
d6_p_i[2]	Input	Positive subLVDS input data lane 6 to subLVDS RX.
d7_p_i[2]	Input	Positive subLVDS input data lane 7 to subLVDS RX.
d8_p_i[2]	Input	Positive subLVDS input data lane 8 to subLVDS RX.
d9_p_i[2]	Input	Positive subLVDS input data lane 9 to subLVDS RX.
d10_p_i[2]	Input	Positive subLVDS input data lane 10 to subLVDS RX.
d11_p_i[2]	Input	Positive subLVDS input data lane 11 to subLVDS RX.
d12_p_i[2]	Input	Positive subLVDS input data lane 12 to subLVDS RX.
d13_p_i[2]	Input	Positive subLVDS input data lane 13 to subLVDS RX.
d14_p_i[2]	Input	Positive subLVDS input data lane 14 to subLVDS RX.
d15_p_i[2]	Input	Positive subLVDS input data lane 15 to subLVDS RX.
drop_line_cnt_i[2:0][3]	Input	Number of dropped lines.
drop_pixel_cnt_i[15:0][3]	Input	Number of dropped pixels.
wc_i[15:0][3]	Input	Word count.
dvalid_o[3]	Output	Data valid detection signal, indicates valid pixel data.
fv_o	Output	Frame valid detection signal, indicates valid frame.
lv_o	Output	Line valid detection signal, indicates a valid line.
pixclk_o	Output	Pixel clock generated from the CLKDIV.
pixdata_o[BUS_WIDTH*LANE_WIDTH][4,5]	Output	Pixel data coming from the parser module. Multi-pixel data bus.
xvs_o[1]	Output	Sony passive readout vertical control signal.
xhs_o[1]	Output	Sony passive readout horizontal control signal.

Notes:

1. *inck_i*, *xvs_o*, and *xhs_o* ports are only available with *Image Sensor Mode = Passive*.
2. *d*N_*p_i* ports are only available if *Number of RX Lanes* > N.
3. BUS_WIDTH depends on *Data Type*. When *Data Type* == RAW10, if *RX Gear* == 8, BUS_WIDTH == 10, else BUS_WIDTH == 20. When *Data Type* == RAW12, if *RX Gear* == 8, BUS_WIDTH == 12, else BUS_WIDTH == 24.
   LANE_WIDTH is equal to *Number of RX Lanes*.
4. The port is available when *Enable APB Interface* == OFF.
5. The port is available when *Enable AXI4-Stream Transmit Interface* == OFF.

4.2. APB Control Interface

Table 4.2. APB Control Interface Ports[1]

Port	Type	Description
apb_pclk_i	Input	An external clock with a maximum frequency of 144 MHz.
apb_presetn_i	Input	APB active low reset.
apb_paddr_i[31:0]	Input	APB address bus.
apb_psel_i	Input	APB select pin to indicate the target is selected and that a data transfer is required.
apb_penable_i	Input	APB enable pin to indicate the second and subsequent cycles of an APB transfer.
apb_pwrite_i	Input	APB read and write signal pin. Indicates APB write access when the signal is high, and APB read access when the signal is low.
apb_pwdata_i[31:0]	Input	APB write data bus. This bus must be driven during write cycles (apb_write_i is high).
apb_prdata_o[31:0]	Output	APB read data bus. The target device drives this bus during read cycles(apb_write_i is low).
apb_pready_o	Output	APB ready signal.
apb_pslverr_o[2]	Output	APB target error signal. Target device asserts this signal to high to indicate a transfer failure.

Notes:

1. These ports are only available when *Enable APB Interface = ON*.
2. This port is available on IP Generation but unused by the design.

4.3. AXI-Stream Transmitter Interface

Table 4.3. AXI-Stream Transmitter Interface Ports[1]

Port	Type	Description
axis_tx_aclk_i	Input	AXI4-Stream Transmit clock with the same frequency as the pixel clock generated from CLKDIV.
axis_tx_aresetn_i	Input	AXI4-Stream Transmit active low reset.
axis_tx_tvalid_o	Output	AXI4-Stream Transmit valid mapped from *dvalid_o* signal from the Native Interface.
axis_tx_tdata_o	Output	AXI4-Stream Transmit data-in mapped from *pixdata_o* signal from the Native Interface.
axis_tx_tuser_o[1:0]	Output	AXI4-Stream Transmit user-defined output mapped from *fv_o* and *lv_o* signals from the Native Interface. Bit 0 – lv_o Bit 1 – fv_o

Note:

1. These ports are only available when *Enable AXI4-Stream Transmit Interface = ON*.

5. Register Description

All registers are accessed through the APB Interface.

Table 5.1. Access Types

Access Types	Behavior on Read Access	Behavior on Write Address
RW	Returns register value	Updates register value
RSVD	Returns 0	Ignores write access

5.1. Configuration Registers

The design maintains configuration registers summarized in the following table.

Table 5.2. Summary of Registers

Offset [3:0]	Register Name	Access	Description
0x000	DROP_PIXEL_CNT	RW	Register for the number of dropped pixels
0x004	DROP_LINE_CNT	RW	Register for the number of dropped lines
0x008	WC	RW	Register for word count

5.1.1. DROP_PIXEL_CNT Register

This register records the number of dropped pixels when APB is enabled.

Table 5.3. DROP_PIXEL_CNT Register

Field	Name	Description	Default
[31:16]	RSVD	Reserved bits	0
[15:0]	drop_pixel_cnt_o	Number of dropped pixels	0

5.1.2. DROP_LINE_CNT Register

This register records the number of dropped lines when APB is enabled.

Table 5.4. DROP_LINE_CNT Register

Field	Name	Description	Default
[31:3]	RSVD	Reserved bits	0
[2:0]	drop_line_cnt_o	Number of dropped lines	0

5.1.3. WC Register

This register records the word count when APB is enabled.

Table 5.5. WC Register

Field	Name	Description	Default
[31:16]	RSVD	Reserved bits	0
[15:0]	wc_o	Number of dropped lines	0

If APB Interface is not enabled, the following internal signals turn to top-level input signals.

- drop_line_cnt_i
- drop_pixel_cnt_i
- wc_i

6. Designing with the IP

This section provides information on how to generate the IP Core using the Lattice Radiant software and how to run simulation and synthesis. For more details on the Lattice Radiant software, refer to the Lattice Radiant Software User Guide.

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

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. To generate the SubLVDS Image Sensor Receiver IP in the Lattice Radiant software, follow these steps:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click **SubLVDS Image Sensor Receiver** under **IP, Audio_Video_and_Image_Processing** category. The **Module/IP Block Wizard** opens as shown in the following figure. Enter values in the **Component name** and the **Create in** fields and click **Next**.

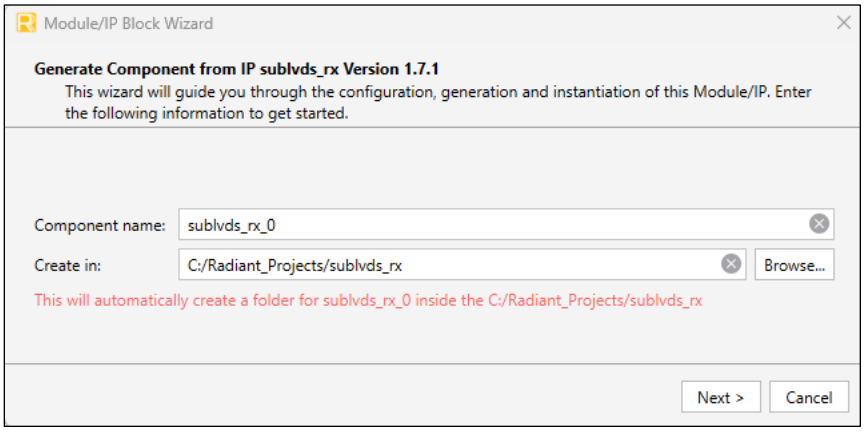

Figure 6.1. Module/IP Block Wizard

3. In the next **Module/IP Block Wizard** window, customize the selected **SubLVDS Image Sensor Receiver IP** using drop-down lists and check boxes. The following figure shows an example configuration of the SubLVDS Image Sensor Receiver IP. For details on the configuration options, refer to the IP Parameter Description section.

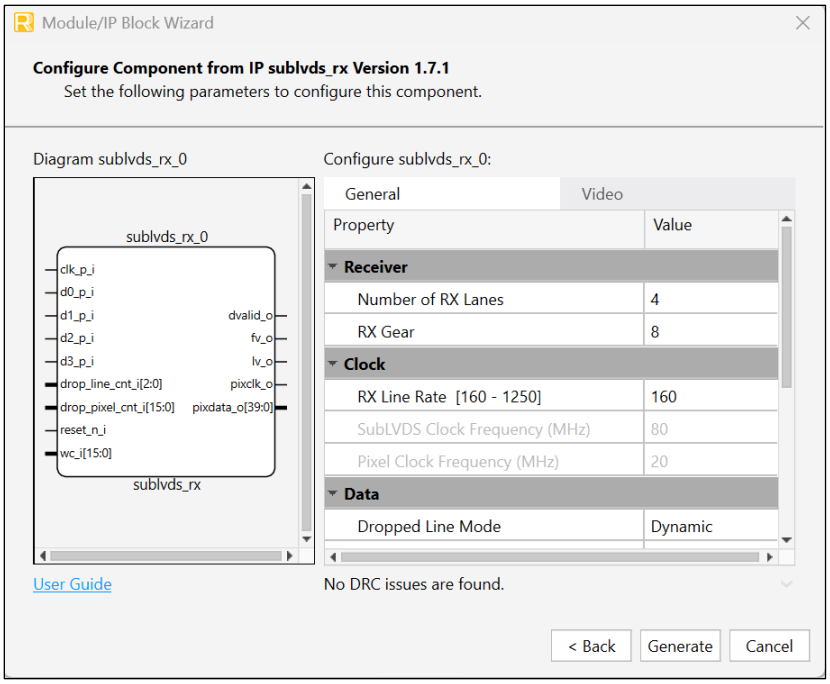

Figure 6.2. IP Configuration

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

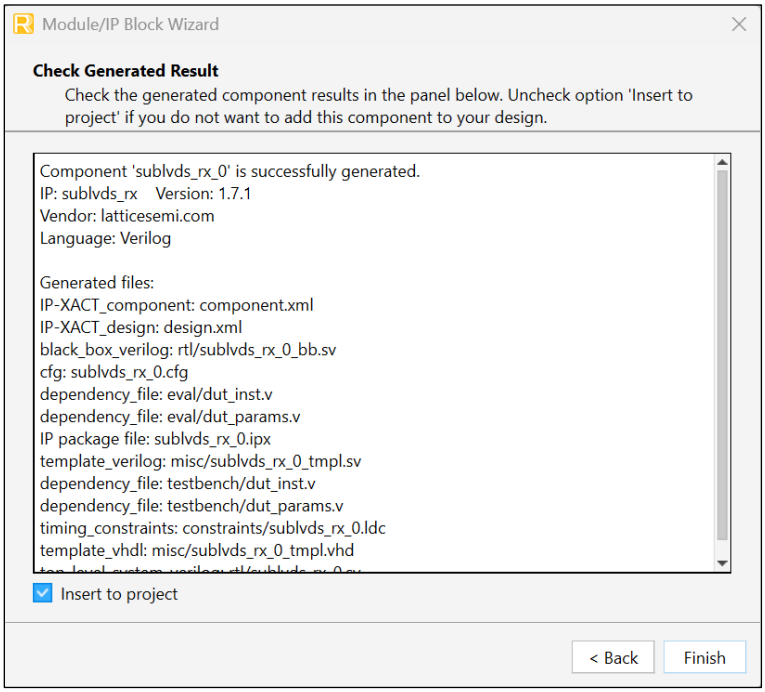

Figure 6.3. Check Generated Result

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 SubLVDS Image Sensor 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 for your complete design. The generated files are listed in the following table.

Table 6.1. Generated File List

Attribute	Description
<Component name>.ipx	This file contains the information on the files associated to 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 misc /<Component name>_tmpl.vhd	These files provide instance templates for the module.

6.2. Design Implementation

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

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

Refer to the relevant sections in the Lattice Radiant Software User Guide for more information on how to create or edit constraints and how to use the Device Constraint Editor.

6.3. Timing Constraints

You need to provide proper timing and physical design constraints to ensure that your design meets the desired performance goals on the FPGA. Add the content of the following IP constraint file to your design constraints: *<IP_Instance_Path>/<IP_Instance_Name>/eval/constraints.pdc*. The constraint file has been verified during IP evaluation with the IP instantiated directly in the top-level module. You can modify the constraints in this file with thorough understanding of the effect of each constraint. To use this constraint file, copy the content of *constraints.pdc* to the top-level design constraint for post-synthesis.

Refer to Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059) for details on how to constraint your design.

6.4. Specifying the Strategy

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

6.5. Running Functional Simulation

You can run functional simulation after the IP is generated.

To run functional simulation, follow these steps:

1. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in the following figure.

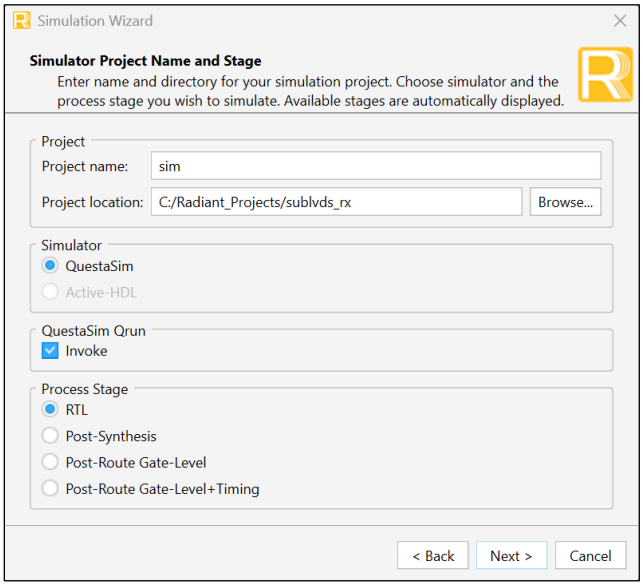

Figure 6.4. Simulation Wizard

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

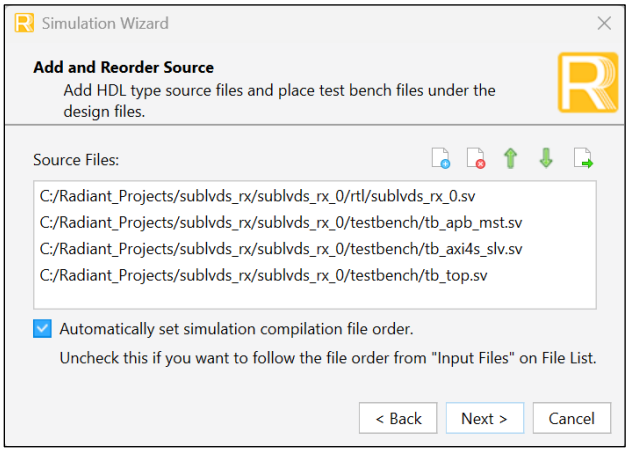

Figure 6.5. Add and Reorder Source

Table 6.2. Testbench Files List

Testbench Files	Description
testbench/tb_top.sv	Top testbench to run loopback test of the generated <Instance Name>.sv
testbench/tb_apb_mst.sv	Testbench to run test on APB Completer Interface.
testbench/tb_axi4s_slv.sv	Testbench to run test on AXI4-Stream Transmitter Interface.

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

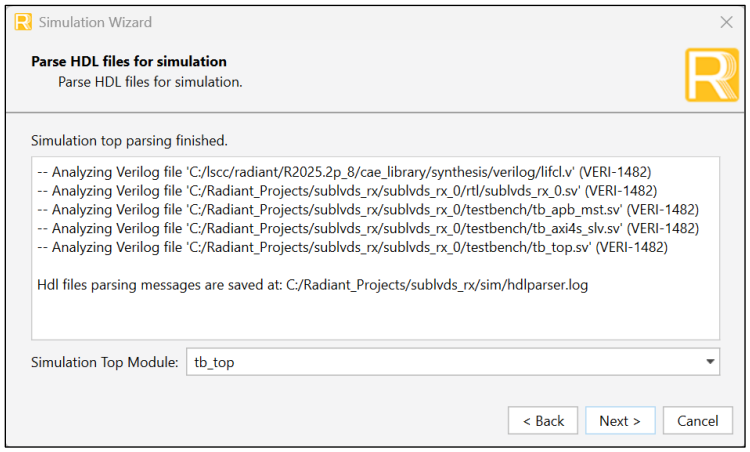

Figure 6.6. Parse HDL Files for Simulation

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

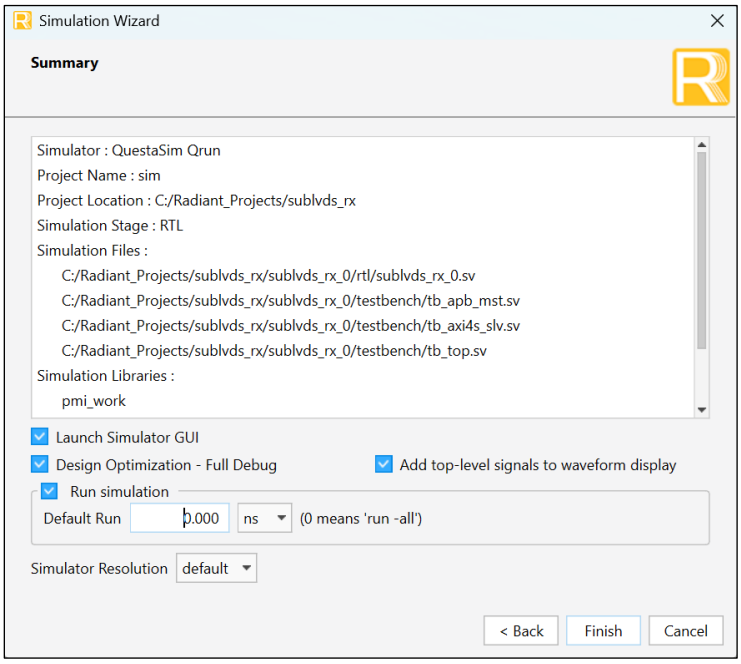

Figure 6.7. Summary

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

6.5.1. Simulation Results

The waveform in the following figure shows an example simulation result.

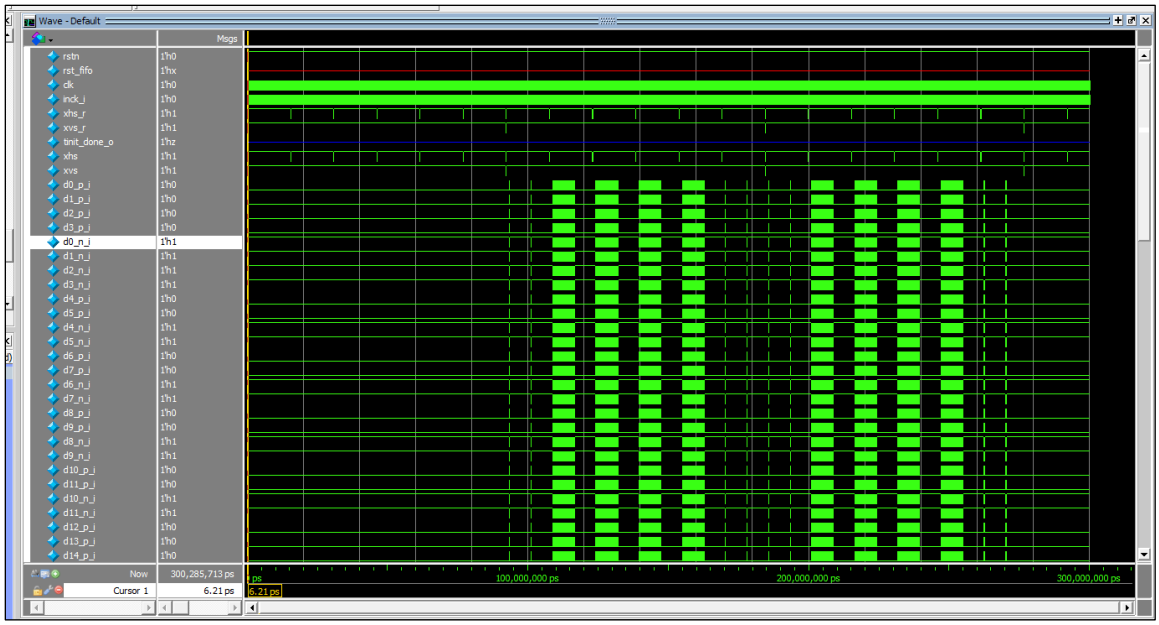

Figure 6.8. Simulation Waveform

The simulation performs a loopback test and compares the received data with the expected data. A successful simulation result with *SIMULATION PASSED* is displayed in the QuestaSim transcript window as shown in the following figure.

```
#          13985629900 : Expected and Received Data are matching (FRAME 1 :: LINE 2)
#             Expected : 0x000000000000
#             Received : 0x000000000000
#
#          13988297100 : Expected and Received Data are matching (FRAME 1 :: LINE 2)
#             Expected : 0x9d09d09d09d0
#             Received : 0x9d09d09d09d0
#
#          14994164900 : End of frame :           1
#
# ------------------------------------------------
# ------------------------------------------------
##### DATA COMPARE STARTED #####
# ------------------------------------------------
# ------------------------------------------------
#  Test fail count :          0
#
# ****SIMULATION PASSED****
# 17994164900 TEST END
```

Figure 6.9. Simulation Result

7. Debugging

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

7.1. Debug Tools

You can use various tools to debug SubLVDS Image Sensor Receiver IP design issues.

7.1.1. Reveal Analyzer

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

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

Before running the Reveal Analyzer, use the Reveal Inserter to add Reveal modules to your design. In these modules, specify the signals to monitor, define the trigger conditions, and set other preferred options. The Reveal Analyzer supports multiple logic analyzer cores using hard 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.

During debug cycles, this tool uses a divide and conquer method to narrow down to problem areas into many small functional blocks to control and monitor the status of each block.

Refer to the Reveal User Guide for Radiant Software for details on how to use the Reveal Analyzer.

7.1.2. QuestaSim

The Siemens® QuestaSim tool is an OEM simulation tool that is closely linked to the Lattice Radiant software environment and can be used to perform functional verification of your design and IP. A proper testbench needs to be written to provide input stimulus to the Device Under Test (DUT) and observe the output signals via the QuestaSim Waveform Viewer to verify the correctness of the IP or design. Lattice provides IP testbench and other simulation files when you generate the IP in Lattice Radiant software. You can use these files 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.

8. Design Considerations

- The dvalid_o signal may be de-asserted during the valid line while waiting for the next valid pixels. You should monitor the dvalid_o signal in conjunction with lv_o to ensure that the downstream logic does not receive invalid data.
- The number of pixels sent to the IP must be aligned with the MIPI requirements.
- To ensure that the input data to the AXI4-Stream transmitter is in sync with the pixclk_o signal, the axis_tx_aclk_i signal must be connected to the pixclk_o signal.
- The Front Dummy and VBLNK period depend on your application or must follow the target specification.

8.1. Limitations

Some IP configurations may fail Static Timing Analysis when using Nexus devices with *RX Line Rate* == 1250 Mbps running in speed grades 7 and 9. The following value is approximate and may very depending on the system-level design:

- Fmax: 107.956 MHz for target frequency of 156.25 MHz (pixclk_o).

Appendix A. Resource Utilization

Table A.1. Device and Tool Tested

—	Value
Software Version	Lattice Radiant software 2025.2
Device Used	LAV-AT-E70-1LFG1156I
Performance Grade	1
Synthesis Tool	Synplify Pro®, April 2025

Note: Some bits are clipped to accommodate the current configuration with the selected device.

Table A.2. SubLVDS-RX Resource Utilization

Number of RX Lanes	RX Gear	Line Rate (Mbps)	Register	LUTs	F_{max}[1]
4	8	900	380	560	clk_p_i = 250 MHz
					pixclk_o = 187.512 MHz
10	8	900	853	1030	clk_p_i = 250 MHz
					pixclk_o = 168.919 MHz
16	8	900	938	2030	clk_p_i = 250 MHz
					pixclk_o = 142.369 MHz

Note:

1. For the given line rate, Fmax is the maximum clock frequency that can be achieved. The maximum frequency may be higher than the values in this table, however, the IP maintains your desired configuration. Fmax is generated using multiple iterations of Place and Route.

Table A.3. Device and Tool Tested

—	Value
Software Version	Lattice Radiant software 2025.1
Device Used	LN2-CT-20ES-1ASG410I
Performance Grade	1
Synthesis Tool	Synplify Pro®, April 2025

Note: Some bits are clipped to accommodate the current configuration with the selected device.

Table A.4. SubLVDS-RX Resource Utilization

Number of RX Lanes	RX Gear	Line Rate (Mbps)	Register	LUTs	F_{max}[1]
4	8	900	380	560	clk_p_i = 250 MHz
					pixclk_o = 168.719 MHz
10	8	900	853	1030	clk_p_i = 250 MHz
					pixclk_o = 180.865 MHz
16	8	900	938	2030	clk_p_i = 250 MHz
					pixclk_o = 152.439 MHz

Note:

1. For the given line rate, Fmax is the maximum clock frequency that can be achieved. The maximum frequency may be higher than the values in this table, however, the IP maintains your desired configuration. Fmax is generated using multiple iterations of Place and Route.

References

- SubLVDS Image Sensor Receiver IP Release Notes (FPGA-RN-02051)
- Lattice Radiant Timing Constraints Methodology (FPGA-AN-02059)
- Avant-E web page
- Avant-G web page
- Avant-X web page
- MachXO5-NX web page
- CrossLink-NX web page
- CertusPro-NX web page
- Certus-NX web page
- Certus-N2 web page
- SubLVDS Image Sensor Receiver IP Core web page
- Lattice Radiant Software web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

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

Revision History

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 2.2, IP v1.7.1, December 2025

Section	Change Summary
All	Added a note on IP version in Quick Facts and *Revision History* sections.
Abbreviations in This Document	Updated list of abbreviations.

Introduction	• Updated Table 1.2. SubLVDS Image Sensor Receiver IP Core Quick Facts as follows:
	  • Added IP version.
	  • Removed earlier IP versions.
	• Updated the Licensing and Ordering Information section.

Functional Description	• Updated that the specific internal signals turn to top-level output signals if AXI4-Stream Transmitter is not enabled in the Custom Transmission section. • Updated Figure 2.10. SubLVDS Image Sensor Receiver IP Core Input Bus Waveform.

IP Parameter Description	• Updated the *RX Line Rate* attribute in Table 3.1. General Attributes.
	• Updated Table 3.2. Video as follows:
	  • Updated the description for the *H_TOTAL* attribute.
	  • Added a note for *H_TOTAL* and *V_H_BLANK* attributes.

Signal Description	Added a note to the apb_pslverr_o signal in Table 4.2. APB Control Interface Ports1.
Register Description	Updated table header from *Offset* to *Offset [3:0]* in Table 5.2. Summary of Registers.

Designing with the IP	• Added a note on IP version in GUI in the Designing with the IP section.
	• Updated the following figures:
	  • Figure 6.1. Module/IP Block Wizard.
	  • Figure 6.2. IP Configuration.
	  • Figure 6.3. Check Generated Result.
	  • Figure 6.4. Simulation Wizard.
	  • Figure 6.5. Add and Reorder Source.
	  • Figure 6.6. Parse HDL Files for Simulation.
	  • Figure 6.7. Summary.

Design Considerations	• Added guidelines in the Design Considerations section. • Updated the section title from *Known Issues* to Limitations, and updated the section.
Resource Utilization	Updated section for the latest software version.
References	Updated references.

Revision 2.1, IP v1.7.0, July 2025

Section	Change Summary

Introduction	• Updated Table 1.2 SubLVDS Image Sensor Receiver IP Core Quick Facts as follows:
	  • Renamed Supported FPGA Families to Supported Devices.
	  • Removed the *Targeted Devices* row.
	  • Added IP version.
	• Updated Table 1.4 Ordering Part Number as follows:
	  • Changed from *Multi-site Perpetual* to *Single Seat Perpetual*.
	  • Updated device name from Lattice Avant-AT-E to Lattice Avant-E, Lattice Avant-AT-G to Lattice Avant-G, and Lattice Avant-AT-X to Lattice Avant-X.

Signal Description	Removed note for *fv_o* and *lv_o* in Table 4.1. Clock Ports.
Designing with the IP	Updated the following figures: • Figure 6.1. Module/IP Block Wizard • Figure 6.2. IP Configuration • Figure 6.3. Check Generated Result

Section	Change Summary
	• Figure 6.4. Simulation Wizard • Figure 6.5. Add and Reorder Source • Figure 6.6. Parse HDL Files for Simulation • Figure 6.7. Summary
Design Considerations	Added this section.
Resource Utilization	Updated resource utilization for the latest software version.

Revision 2.0, IP v1.6.0, December 2024

Section	Change Summary
Abbreviations in This Document	Updated abbreviations.

Introduction	• Updated Table 1.2 SubLVDS Image Sensor Receiver IP Core Quick Facts as follows:
	  • Added support for LFD2NX-9, LFD2NX-28, and LN2-CT-20 devices.
	  • Added IP changes.
	  • Updated IP version.
	• Added the IP Support Summary section.
	• Updated Table 1.4 Ordering Part Number as follows:
	  • Added OPN for Certus-N2 devices.
	  • Changed from *Single Machine Annual* to *Single Seat Annual*.
	• Removed the *IP Validation Summary* section.

Resource Utilization	Added resource utilization for the LN2-CT-20-1CBG484C device.
References	Added links to the Certus-N2 web page and IP release notes.

Revision 1.9, September 2024

Section	Change Summary
All	Renamed document from *SubLVDS Image Sensor Receiver IP Core* to *SubLVDS Image Sensor Receiver IP*.

Introduction	• Reworked section 1 *Introduction* and moved to subsection 1.1 Overview of the IP.
	• Moved subsection 1.1 *Quick Facts* to subsection 1.2 Quick Facts.
	• Moved subsection 1.2 Features to subsection 1.3 Features.
	• Reworked subsection *3.1 Licensing the IP* and section *4 Ordering Part Number* and renamed to subsection 1.4 Licensing and Ordering Information.
	• Added the following subsections:
	  • 1.5 IP Validation Summary
	  • 1.6 Minimum Device Requirements
	• Reworked subsection 1.3 *Conventions* and renamed to subsection 1.7 Naming Conventions.

Functional Description	• Moved subsection 2.1 *Overview* and renamed to subsection 2.1 IP Architecture Overview. • Moved subsection 2.4.1 *Clock, Reset and Initialization* to subsection 2.2 Clocking and subsection 2.3 Reset. • Reworked subsection 2.6 *AXI4-Stream Transmit Interface* and renamed to subsection 2.4 User Interfaces. • Reworked subsection 2.4.2 *SubLVDS Image Sensor Receiver IP Core Submodules* and moved to subsection 2.5 SubLVDS Image Sensor Receiver IP Core Submodules. • Reworked subsection 2.7 *Timing Specifications* and renamed to subsection 2.6 Timing Specifications. • Reworked subsection 2.8 *Sample Configurations* and renamed to subsection 2.7 Sample Configurations.
IP Parameter Description	Reworked subsection 2.3 *Attribute Summary* and renamed to section 3 IP Parameter Description.

Section	Change Summary
Signal Description	Reworked subsection 2.2 *Signal Description* and moved to section 4 Signal Description.
Register Description	Moved subsection 2.5 *Register Descriptions* to section 5 Register Description.

Designing with the IP	• Reworked subsection 3.2 *Generation and Synthesis* and renamed to subsection 6.1 Generating and Instantiating the IP.
	• Added the following subsections:
	  • 6.2 Design Implementation
	  • 6.4 Specifying the Strategy
	• Renamed subsection 3.4 *Constraining the IP* and moved to subsection 6.3 Timing Constraints.
	• Reworked subsection 3.3 *Running Functional Simulation* and moved to subsection 6.5 Running Functional Simulation.

Debugging	Added this section.
Appendix A. Resource Utilization	Updated for the latest software version.
References	Updated references.

Revision 1.8, December 2023

Section	Change Summary
All	• Updated title from SubLVDS Image Sensor Receiver IP Core – Lattice Radiant Software to *SubLVDS Image Sensor Receiver IP Core*. • Performed minor editorial fixes.
Disclaimers	Updated this section.
Inclusive Language	Added this section.
Introduction	Updated *LAV-AT-500E* to *LAV-AT-E70* and added devices *LAV-AT-G70* and *LAV-AT-X70* in Table 1.1. SubLVDS Image Sensor Receiver IP Core Quick Facts.
IP Generation and Evaluation	• Fixed broken link in the Licensing the IP section. • Updated the Constraining the IP section.
Ordering Part Number	Updated Table 4.1 with Avant-AT-E, Avant-AT-G, and Avant-AT-X part numbers.
Appendix A. Resource Utilization	• Updated device from *LAV-AT-500E-3LFG1156C* to *LAV-AT-E70-3LFG1156C* in Table A.2. SubLVDS-RX Resource Utilization. • Updated note below Table A.2. SubLVDS-RX Resource Utilization.
References	Added links to the Avant-G and Avant-X web pages.

Revision 1.7, September 2023

Section	Change Summary
All	• Changed all instances of *Master* to *Active*. • Changed all instances of *Slave* to *Passive*.

Introduction	• Table 1.1. SubLVDS Image Sensor Receiver IP Core Quick Facts:
	  • added *MachXO5-NX* to Supported FPGA Families;
	  • added *LFMXO5-25*, *LFMXO5-55T*, *LFMXO5-100T*, and *LIFCL-33* to Targeted Devices;
	  • added *APB Interface* and *AXI4-Stream Transmit Interface* to Supported User Interface;
	  • added *IP Core v1.4.x - Lattice Radiant Software 2023.1* to Lattice Implementation.
	• In the Features section:
	  • removed the *1 channel* feature;
	  • added *Supports APB Interface for register access and AXI4-Stream Transmit Interface*;
	  • removed *However, this IP Core does not support configuration through registers*.

Section	Change Summary

Functional Description	• Updated Figure 2.1. SubLVDS Image Sensor Receiver IP Core Top Level Block Diagram;
	• Table 2.1. SubLVDS Image Sensor Receiver IP Core Signal Description:
	  • added APB *Completer Interface Ports* and *AXI4-Stream Transmitter Interface Ports*;
	  • added table notes 2 to 5.
	• Table 2.2. Attributes Table:
	  • added the attributes *Enable APB Interface* and *Enable AXI4-Stream Transmit Interface* under Miscellaneous;
	  • added the table note on inclusive language.
	• Added the attributes *Enable APB Interface* and *Enable AXI4-Stream Transmit Interface* under Miscellaneous in Table 2.3. Attributes Description.
	• Updated Figure 2.2. Clock Domain Crossing Block Diagram.
	• Changed the caption of Figure 2.3 to *SubLVDS Image Sensor Receiver IP Core Detailed Block Diagram (APB and AXI4-Stream Transmitter Interfaces not enabled)*.
	• Removed the previous Figure 2.4 Deserializer of 1:8 Gearing Block Diagram, Figure 2.5 Deserializer of 1:16 Gearing Block Diagram, and Figure 2.6 SubLVDS Word Alignment Block Diagram.
	• Added Figure 2.4. SubLVDS Image Sensor Receiver IP Core Detailed Block Diagram (APB Interface Enabled), Figure 2.5. SubLVDS Image Sensor Receiver IP Core Detailed Block Diagram (AXI4-Stream Transmitter Interface Enabled), and Figure 2.6. SubLVDS Image Sensor Receiver IP Core Detailed Block Diagram (APB and AXI4-Stream Interfaces Enabled).
	• Added the Register Descriptions and AXI4-Stream Transmit Interface sections.

IP Generation and Evaluation	• Updated Figure 3.2. Configure Block of SubLVDS Image Sensor Receiver IP Core, Figure 3.3. Check Generated Result, and Figure 3.5. Adding and Reordering Source.
	• Added Table 3.2. Testbench Files List.
	• Added the Constraining the IP section.
	• IP Evaluation:
	  • changed the section name to IP Evaluation;
	  • updated the section content to reflect the section name change;
	  • updated the steps for changing the IP evaluation capability setting to *Project > Active Strategy > Bitstream > IP Evaluation*.

Ordering Part Number	Updated the OPNs for the SubLVDS Image Sensor Receiver IP Core.
References	Updated this section, added links to Lattice Radiant Software Web Page, Lattice Avant-E FPGA Web Page, MachXO5-NX FPGA Web Page, and Lattice Insights for Training Series and Learning Plans.
Appendix A. Resource Utilization	Updated Table A.2. SubLVDS-RX Resource Utilization.

Revision 1.6, March 2023

Section	Change Summary
Disclaimers	Updated this section.

Introduction	• In Table 1.1. SubLVDS Image Sensor Receiver IP Core Quick Facts:
	  • added LAV-AT-500E to the Targeted Devices;
	  • updated Lattice Implementation.
	• In Features, added *12*, *14* and *16* to the data lanes supported by the SubLVDS Image Sensor Receiver IP Core.

Functional Description	• Updated Figure 2.1. SubLVDS Image Sensor Receiver IP Core Top Level Block Diagram.
	• In Table 2.1. SubLVDS Image Sensor Receiver IP Core Signal Description:
	  • removed the complementary ports: d0_n_i, d1_n_i, d2_n_i, d3_n_i, d4_n_i, d5_n_i, d6_n_i, d7_n_i, d8_n_i, d9_n_i;
	  • added the following ports: d10_p_i, d11_p_i, d12_p_i, d13_p_i, d14_p_i, d15_p_i.

Section	Change Summary
	• In Table 2.2. Attributes Table, added *12*, *14*, and *16* to the selectable values for the *Number of RX Lanes*. • Updated Figure 3.3. Check Generated Result.
IP Generation and Evaluation	• Updated Figure 3.2. Configure Block of SubLVDS Image Sensor Receiver IP Core. • Updated Figure 3.3. Check Generated Result.
Ordering Part Number	Added the ordering part numbers for Lattice Avant.
Appendix A. Resource Utilization	• In Table A.1. Device and Tool Tested, updated the *Lattice Radiant Software Version*, *Device Used*, *Performance Grade* and *Synthesis Tool* for the LAV-AT-500E-3LFG1156C device. • In Table A.2. SubLVDS-RX Resource Utilization, updated the values for the *Number of RX Gears*, *RX Gear*, *Line Rate*, *Synthesis Tool*, *Register*, *LUTs*, and *Fmax*.

Revision 1.5, June 2021

Section	Change Summary
Introduction	• Updated Table 1.1. SubLVDS Image Sensor Receiver IP Core Quick Facts. • Revised Supported FPGA Families • Revised Targeted Devices • Revised Lattice Implementation
IP Generation and Evaluation	• In the Hardware Evaluation section, replaced specific device with *Lattice FPGA devices built on the Lattice Nexus platform*.
Ordering Part Number	Added part numbers.
References	Added reference to the CertusPro-NX web page.

Revision 1.4, December 2020

Section	Change Summary
Introduction	Updated Table 1.1. Modified Lattice Implementation details.
Functional Description	• Updated RX Line Rate selectable values in Table 2.2. Attributes Table. • Updated Line Rate information in Table 2.3. Attributes Description.
IP Generation and Evaluation	Updated Figure 3.2. Configure Block of SubLVDS Image Sensor Receiver IP Core.
References	Updated this section. Added references to product web pages.
All	Updated Lattice Radiant Software User Guide references.

Revision 1.3, June 2020

Section	Change Summary
All	Updated Lattice Radiant software user guide references to version 2.1 across the document.
Introduction	Added support for Certus-NX in Table 1.1.
Ordering Part Number	Updated this section.
Appendix A. Resource Utilization	Added this section.

Revision 1.2, April 2020

Section	Change Summary
Functional Description	Corrected maximum line rate in Table 2.2.

Revision 1.1, February 2020

Section	Change Summary
Introduction	• Updated Table 1.1 to add LIFCL-17 as targeted device.

Section	Change Summary
	• Removed section 1.3.2. Data Ordering and Data Types
Functional Description	• Added pixclk_o port and note to Table 2.1. • Revised V_TOTAL values in Table 2.2. • Changed column heading to Description and updated descriptions of Dropped Pixel Count and Word Count attributes in Table 2.3. • Added Sample Configurations section.
IP Generation and Evaluation	Corrected interface item to *Check Generated Result*.
Appendix A.	Added table reference in introductory paragraph.

Revision 1.0, December 2019

Section	Change Summary
All	Initial release

www.latticesemi.com