

SMPTE UHD-SDI Receiver Subsystem v1.0

LogiCORE IP Product Guide

Vivado Design Suite

PG290 October 4, 2017

Table of Contents

IP Facts

Chapter 1: Overview

Introduction	5
Subcore Details	5
Applications	7
Unsupported Features.....	7
Licensing.....	7

Chapter 2: Product Specification

Standards	8
Performance.....	8
Resource Utilization.....	9
Port Descriptions	9
Register Space	13

Chapter 3: Designing with the Core

General Design Guidelines	27
Clocking.....	28
Resets	29

Chapter 4: Design Flow Steps

Customizing and Generating the Core	31
Core Configuration Tab	32
Application Example Design Tab.....	33
Constraining the Core	34
Simulation	35
Synthesis and Implementation	35

Chapter 5: Example Design

ZCU106 UHD-SDI Pass-Through Example Design.....	36
---	----

Appendix A: Verification, Compliance, and Interoperability

Interoperability	60
------------------------	----

Hardware Validation 61
Video Resolutions 61

Appendix B: Debugging

Finding Help on Xilinx.com 75
Debug Tools 76
Hardware Debug 77
Interface Debug 80

Appendix C: Additional Resources and Legal Notices

Xilinx Resources 82
Documentation Navigator and Design Hubs 82
References 83
Revision History 83
Please Read: Important Legal Notices 83

Introduction

The Society of Motion Picture and Television Engineers (SMPTE) UHD-SDI receiver subsystem implements a SDI receive interface in accordance to the serial digital interface (SDI) family of standards. The subsystem receives video from a native SDI interface and generates AXI-4 Stream Video. The subsystem allows fast selection of the top level parameters and automates most of the lower level parameterization. The AXI4-Stream video interface allows a seamless interface to other AXI4-Stream-based subsystems.

Features

- Support for 2 pixel per sample.
- 10-bit per color component.
- Supports YUV 4:2:2 color space.
- AXI4-Lite interface for register access to configure different subsystem options.
- Standards compliance:
 - SMPTE ST 259: SD-SDI at 270 Mb/s.
 - SMPTE RP 165: EDH for SD-SDI.
 - SMPTE ST 292: HD-SDI at 1.485 Gb/s and 1.485/1.001 Gb/s.
 - SMPTE ST 372: Dual Link HD-SDI.
 - SMPTE ST 424: 3G-SDI with data mapped by any ST 425-x mapping at 2.97 Gb/s and 2.97/1.001 Gb/s.
 - SMPTE ST 2081-1: 6G-SDI with data mapped by any ST 2081-x mapping at 5.94 Gb/s and 5.94/1.001 Gb/s.
 - SMPTE ST 2082-1: 12G-SDI with data mapped by any ST 2082-x mapping at 11.88 Gb/s and 11.88/1.001 Gb/s.
 - Dual link and quad link 6G-SDI and 12G-SDI are supported by instantiating two or four UHD-SDI Receiver subsystems.
 - SMPTE ST 352: Payload ID packets are fully supported.

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family ⁽¹⁾	UltraScale+™ Families (GTHE4) Zynq® UltraScale+ MPSoC (GTHE4)
Supported User Interfaces	AXI4-Lite, AXI4-Stream
Resources	Performance and Resource Utilization web page
Provided with Core	
Design Files	RTL
Example Design	Verilog
Test Bench	Not provided
Constraints File	XDC
Simulation Model	Not provided
Supported S/W Driver ⁽²⁾	Standalone and Linux
Tested Design Flows⁽³⁾	
Design Entry	Vivado® Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide .
Synthesis	Vivado Synthesis
Support	
Provided by Xilinx @ www.xilinx.com/support	

Notes:

1. For a complete list of supported devices, see the Vivado IP catalog.
2. Standalone driver details can be found in the SDK directory (<install_directory>/doc/usenglish/xilinx_drivers.htm). Linux OS and driver support information is available from <http://www.wiki.xilinx.com/Xilinx+V4L2+SDI+Rx+driver>.
3. For the supported versions of the tools, see the [Xilinx Design Tools: Release Notes Guide](#).

Overview

Introduction

The SMPTE UHD-SDI Receiver Subsystem allows you to quickly create systems based on SMPTE SDI protocols. It accepts native SDI stream and outputs AXI-4 Video stream by using Xilinx® transceivers as physical layer. The top-level customization parameters select the required hardware blocks needed to build the subsystem. [Figure 1-1](#) shows the subsystem architecture.

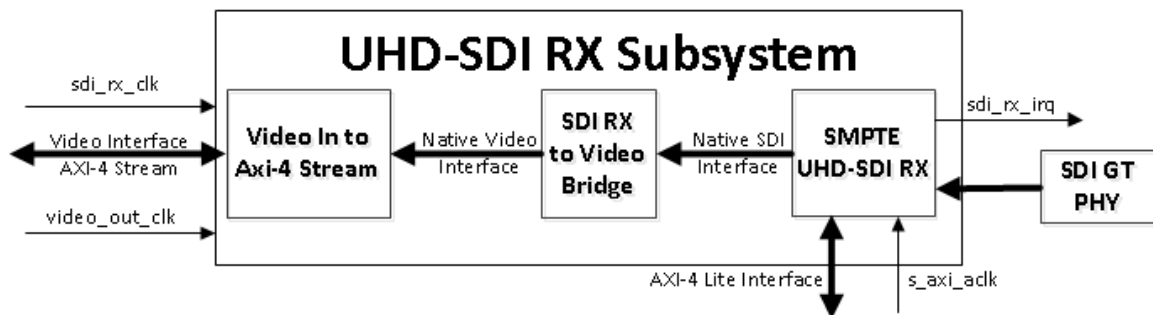


Figure 1-1: Receiver Subsystem Architecture

The subsystem consists of the following subcores:

- SMPTE UHD-SDI RX
- SDI RX to Video Bridge
- Video In to AXI-4 Stream

Subcore Details

SMPTE UHD-SDI Receiver

The SMPTE UHD-SDI receiver core receives multiplexed native SDI data streams and generates non-multiplexed 10-bit SDI data stream. See the *SMPTE UHD-SDI Product Guide* [Ref 8] for details.

SDI RX to Video Bridge

The LogiCORE IP SDI RX to Video Bridge core is designed to interface from an SDI receiver output of the SMPTE SDI core to the video input of the Video In to AXI4-Stream core. The input is an SDI virtual interface that has one to eight 10-bit data streams with embedded synchronization. The output is video data with explicit synchronization signals. This core extracts synchronization signals, reformats the video data, and provides clock enables.

Figure 1-2 shows the top level bridge architecture.

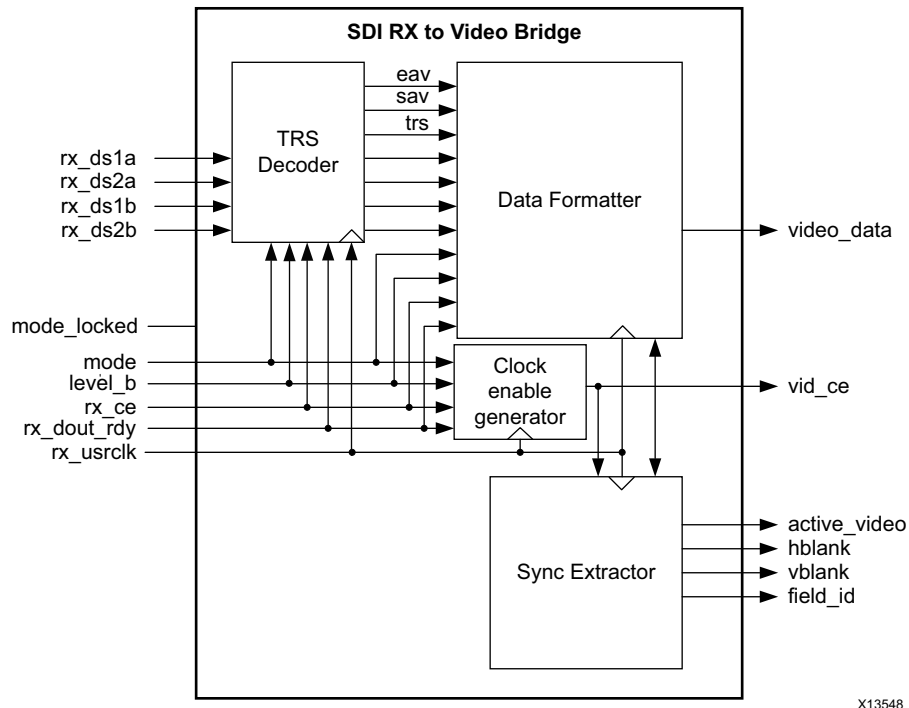


Figure 1-2: Top-Level Block Diagram of SDI RX to Video Bridge

The core extracts embedded synchronization signals from the SDI data stream. It supports SD-SDI, HD-SDI, 3G-SDI Level A, 3G-SDI Level B, 6G-SDI and 12G-SDI with Y, Cb, Cr data format at 10 bits per component. For 3G-SDI Level B, it automatically reorders two lines of parallel data to sequential lines of video data out. It supports both interlaced and progressive line standards.

Video In to AXI-4 Stream

The Video In to AXI-4 Stream core act as an interface from a video source (clocked parallel video data with synchronization signals - active video with either syncs, blanks or both) to the AXI4-Stream Video Protocol Interface. See *Video In to AXI-4 Stream LogiCORE IP Product Guide* [Ref 9] for details.

Applications

- Professional broadcast cameras
- Professional digital video recorders
- Professional video processing equipment
- Medical imaging

Unsupported Features

- 16-way data stream interleaving is not supported.
- YUV 4:2:2 format is supported. All other formats are not supported due to SDI RX bridge.

Licensing

The SMPTE UHD-SDI Receiver Subsystem is provided at no additional cost with the Xilinx Vivado Design Suite under the terms of the [Xilinx End User License](#). Information about this and other Xilinx LogiCORE IP modules is available at the [Xilinx Intellectual Property](#) page. For information about pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your [local Xilinx sales representative](#).

Product Specification

Standards

The core supports the following SMPTE standards:

- SMPTE ST 259: SD-SDI at 270 Mb/s
- SMPTE RP 165: EDH for SD-SDI
- SMPTE ST 292: HD-SDI at 1.485 Gb/s and 1.485/1.001 Gb/s
- SMPTE ST 372: Dual Link HD-SDI (by instantiation of two UHD-SDI cores)
- SMPTE ST 424: 3G-SDI with data mapped by any ST 425-x mapping at 2.97 Gb/s and 2.97/1.001 Gb/s
- SMPTE ST 2081-1: 6G-SDI with data mapped by any ST 2081-x mapping at 5.94 Gb/s and 5.94/1.001 Gb/s (including multi-link 6G-SDI)
- SMPTE ST 2082-1: 12G-SDI with data mapped by any ST 2082-x mapping at 11.88 Gb/s and 11.88/1.001 Gb/s (including multi-link 12G-SDI)

Dual link and quad link 6G-SDI and 12G-SDI are supported by instantiating two or four UHD-SDI cores.

- SMPTE ST 352: Payload ID packets are fully supported.
-

Performance

Maximum Frequencies

In 12G-SDI mode, the maximum frequency of the RX clock is 297 MHz. In 6G-SDI, 3G-SDI, and SD-SDI modes, the maximum frequency of the RX clock is 148.5 MHz. In HD-SDI mode, the maximum frequency of the RX clock is 74.25 MHz.

Resource Utilization

For full details about performance and resource utilization, visit the [Performance and Resource Utilization web page](#).

Port Descriptions

The SMPTE UHD-SDI RX Subsystem I/O signals are described in [Table 2-1](#).

Table 2-1: Port Descriptions

Signal	Direction	Description
AXI-4 Lite Interface Signals (when Enable AxILite Interface option is selected)		
s_axi_aclk	Input	AXI-4 Lite clock
s_axi_arstn	Input	AXI-4 Lite reset. Active-Low
S_AXI_CTRL*		AXI4-Lite interface, defined in the <i>Vivado Design Suite: AXI Reference Guide</i> (UG1037) [Ref 13].
Video-Over-AXIS Interface Signals (when Enable Vid-Over-AXI4S Interface option is selected)		
video_out_clk	Input	Video output clock
video_out_arstn	Input	Video output active-Low reset.
VIDEO_OUT_tdata[63:0]	Output	Video input data for carrying YUV 4:2:2 video with 10 bpc. (for details refer to PG044, AXI4-Stream Data Interface Signal Descriptions)
VIDEO_OUT_tlast	Output	AXI4-Stream TLAST. End of Line
VIDEO_OUT_tready	Input	AXI4-Stream TREADY.
VIDEO_OUT_tuser	Output	AXI4-Stream TUSER. Start of Frame
VIDEO_OUT_tvalid	Output	AXI4-Stream TVALID. Active video data enable
fid	Output	Field ID
S_AXIS_STS_SB_RX Interface Signals		
S_AXIS_STS_SB_RX_tready	Output	Core Ready
S_AXIS_STS_SB_RX_tvalid	Input	Data valid
S_AXIS_STS_SB_RX_tdata[31:0]	Input	Sideband signal information from transceiver block
S_AXIS_RX Interface Signals		
sdi_rx_clk	Input	SMPTE SDI RX Core clock
sdi_rx_rstn	Input	Active low reset
S_AXIS_RX_tready	Input	SMPTE SDI RX Core ready
S_AXIS_RX_tvalid	Output	Data valid

Table 2-1: Port Descriptions (Cont'd)

Signal	Direction	Description
S_AXIS_RX_tdata[n-1:0]	Output	n is varies with SDI standard selection. n=40 for 6G-SDI and 12G-SDI n=20 for 3G-SDI
S_AXIS_RX_tuser[31:0]	Output	TUSER Information
M_AXIS_CTRL_SB_RX Interface Signals		
M_AXIS_CTRL_SB_RX__tready	Input	Core Ready
M_AXIS_CTRL_SB_RX_tvalid	Output	Data valid
M_AXIS_CTRL_SB_RX_tdata[31:0]	Output	Sideband signal information from transceiver block
Interrupt Signal		
sdi_rx_irq	Output	SMPTE UHD-SDI RX core interrupt
SDI_TS_DET_OUT Interface Signals⁽¹⁾		
SDI_TS_DET_OUT_rx_t_locked	Output	This output is High when the transport detection function in the receiver has identified the transport format of the SDI signal. (i.e. transport locked)
SDI_TS_DET_OUT_rx_t_family[3:0]	Output	This output indicates which family of video signals is being used as the transport of the SDI interface. This output is only valid when rx_t_locked is High. This port does not necessarily identify the video format of the picture being transported. It only identifies the transport characteristics.
SDI_TS_DET_OUT_rx_t_rate[3:0]	Output	This output indicates the frame rate of the transport. This is not necessarily the same as the frame rate of the actual picture. This output is only valid when rx_t_locked is High.
SDI_TS_DET_OUT_rx_t_scan	Output	This output indicates whether the transport is interlaced (Low) or progressive (High). This is not necessarily the same as the scan mode of the actual picture. This output is only valid when rx_t_locked is High.

Table 2-1: Port Descriptions (Cont'd)

Signal	Direction	Description
SMPTE UHD-SDI RX Core signals⁽¹⁾ (Enabled only when 'Enable Vid-Over-AXI4S Interface' option is not selected)		
sdi_rx_ctrl[63:0]	Input	Bit0: module_enable; Bit1: not used; Bit3~bit2: Reserved; Bit4: rx_frame_en; Bit5: rx_mode_detect_en; Bit6: rx_edh_clr_errcnt Bit7: Reserved; Bit13~bit8: rx_mode_enable: Bit8: enable HD-SDI mode; Bit9: enable SD-SDI mode; Bit10: enable 3G-SDI mode; Bit11: enable 6G-SDI mode; Bit12: enable 12G-SDI 11.88 Gb/s mode Bit13: enable 12G-SDI 11.88/1.001 Gb/s mode. Bit15~bit14: Reserved; Bit18~bit16: rx_forced_mode: 000-HD;001-SD;010-3G;100-6G; 101-12G 11.88 Gb/s; 110-12G 11.88/1.001 Gb/s; Bit31~bit19: Reserved; Bit47~bit32: rx_edh_errcnt_en[15:0]; Bit63~bit48: Reserved;
ST352_DATA_OUT_rx_st352_0[31:0]	Output	ST352 data for channel 0
ST352_DATA_OUT_rx_st352_1[31:0]	Output	ST352 data for channel 1
ST352_DATA_OUT_rx_st352_2[31:0]	Output	ST352 data for channel 2
ST352_DATA_OUT_rx_st352_3[31:0]	Output	ST352 data for channel 3
ST352_DATA_OUT_rx_st352_0_valid	Output	ST352 data valid for channel 0
ST352_DATA_OUT_rx_st352_1_valid	Output	ST352 data valid for channel 1
ST352_DATA_OUT_rx_st352_2_valid	Output	ST352 data valid for channel 2
ST352_DATA_OUT_rx_st352_3_valid	Output	ST352 data valid for channel 3
SDI_DS_OUT_ds1[9:0]	Input	SDI data stream 1
SDI_DS_OUT_ds2[9:0]	Input	SDI data stream 2
SDI_DS_OUT_ds3[9:0]	Input	SDI data stream 3
SDI_DS_OUT_ds4[9:0]	Input	SDI data stream 4
SDI_DS_OUT_ds5[9:0]	Input	SDI data stream 5
SDI_DS_OUT_ds6[9:0]	Input	SDI data stream 6
SDI_DS_OUT_ds7[9:0]	Input	SDI data stream 7

Table 2-1: Port Descriptions (Cont'd)

Signal	Direction	Description
SDI_DS_OUT_ds8[9:0]	Input	SDI data stream 8
SDI_DS_OUT_ln_num_1[10:0] to SDI_DS_OUT_ln_num_8	Input	SDI data stream line number
SDI_DS_OUT_rx_ce_out	Output	Clock enable
SDI_DS_OUT_rx_active_streams	Output	This port indicates the number of data streams that are active for the current video format being received. The number of active data streams is $2^{\text{active_streams}}$. 000: 1 active stream 001: 2 active streams 010: 4 active streams 011: 8 active streams
SDI_DS_OUT_rx_mode_locked	Output	When this output is Low, the receiver is actively searching for the SDI mode that matches the input data stream. During this time, the rx_mode_locked output port changes frequently. When the receiver locks to the correct SDI mode, the rx_mode_locked output goes High.
SDI_DS_OUT_rx_eav	Output	This output is asserted High when the XYZ word of an EAV is present on the data stream output ports.
SDI_DS_OUT_rx_sav	Output	This output is asserted High when the XYZ word of a SAV is present on the data stream output ports.
SDI_DS_OUT_rx_trs	Output	This output is asserted High while the four consecutive words of any EAV or SAV are present on the data stream output ports, from the 3FF word through the XYZ word.
SDI_DS_OUT_rx_mode_hd	Output	High when RX is locked in HD-SDI mode
SDI_DS_OUT_rx_mode_sd	Output	High when RX is locked in SD-SDI mode
SDI_DS_OUT_rx_mode_3g	Output	High when RX is locked in 3G-SDI mode
SDI_DS_OUT_rx_mode_6g	Output	High when RX is locked in 6G-SDI mode
SDI_DS_OUT_rx_mode_12g	Output	High when RX is locked in 12G-SDI mode
SDI_DS_OUT_rx_level_b_3g	Output	IN 3G-SDI mode, this output is asserted High when the input signal is level B and Low when it is level A. This output is only valid when rx_mode_3g is High.

Table 2-1: Port Descriptions (Cont'd)

Signal	Direction	Description
SDI_DS_OUT_sdi_mode[2:0]	Output	This output port indicates the current SDI mode of the receiver: 000 = HD 001 = SD 010 = 3G 100 = 6G 101 = 12G 1000/1000 110 = 12G 1000/1001 When the receiver is not locked, the sdi_mode port changes values as the receiver searches for the correct SDI mode. During this time, the rx_mode_locked output is Low. When the receiver detects the correct SDI mode, the rx_mode_locked output goes High.
sdi_rx_err[31:0]	Output	Bit15~bit0: rx_crc_err_ds16 to rx_crc_err_ds1; Bit31~bit16: Reserved;

Notes:
1. Refer the Table 2-2 of *SMPTE UHD-SDI Product Guide* (PG205) [Ref 8] for more detailed signals descriptions.

Register Space

This section details registers available in the SMPTE UHD-SDI RX Subsystem. SMPTE UHD-SDI RX core is given an address space of 64K.

SMPTE UHD-SDI RX Registers

The SMPTE UHD-SDI RX registers are available when Enable AxiLite Interface is selected in Vivado IDE. The UHD-SDI RX IP core register space is shown in Table 2-2.



IMPORTANT: This memory space must be aligned to an AXI word (32-bit) boundary.

Endianness

All registers are in little endian format as shown in Figure 2-1.

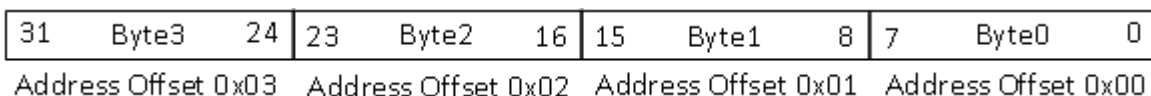


Figure 2-1: 32-bit Little Endian Example

Table 2-2: UHD-SDI RX IP Core Register Space

Offset	Name	Width	Access	Description
0x00	RST_CTRL	32-bit	R/W	Enable and soft reset controls for the IP core
0x04	MODULE_CTRL	32-bit	R/W	Module control register
0x08	RESERVED	32-bit	N/A	N/A
0x0C	GLBL_IER	32-bit	R/W	Global interrupt enable register
0x10	ISR	32-bit	R/W1C	Interrupt status register
0x14	IER	32-bit	R/W	Interrupt enable register
0x18	RX_ST352_VALID	32-bit	RO	ST352 packet valid indication
0x1C	RX_ST352_DATA_DS1	32-bit	RO	Data stream 1 ST352 packet data
0x20	RX_ST352_DATA_DS3	32-bit	RO	Data stream 3 ST352 packet data
0x24	RX_ST352_DATA_DS5	32-bit	RO	Data stream 5 ST352 packet data
0x28	RX_ST352_DATA_DS7	32-bit	RO	Data stream 7 ST352 packet data
0x2C	RX_ST352_DATA_DS9	32-bit	RO	Data stream 9 ST352 packet data
0x30	RX_ST352_DATA_DS11	32-bit	RO	Data stream 11 ST352 packet data
0x34	RX_ST352_DATA_DS13	32-bit	RO	Data stream 13 ST352 packet data
0x38	RX_ST352_DATA_DS15	32-bit	RO	Data stream 15 ST352 packet data
0x3C	VERSION	32-bit	RO	Version Register
0x40	SS_CONFIG	32-bit	RO	IP core Configuration
0x44	MODE_DET_STS	32-bit	RO	Mode detect status
0x48	TS_DET_STS	32-bit	RO	Transport Stream detect status
0x4C	RX_EDH_STS	32-bit	RO	EDH check status
0x50	RX_EDH_ERRCNT_EN	32-bit	R/W	Enable EDH error count
0x54	RX_EDH_ERRCNT	32-bit	RO	RX EDH error count
0x58	RX_CRC_ERR	32-bit	RO	RX CRC error indication
0x5C	VIDEO_LOCK_WINDOW	32-bit	R/W	Video lock window
0x60	RESERVED	32-bit	N/A	N/A

Table 2-2: UHD-SDI RX IP Core Register Space (Cont'd)

Offset	Name	Width	Access	Description
0x64	RESERVED	32-bit	N/A	N/A
0x68	RESERVED	32-bit	N/A	N/A
0x6C	RESERVED	32-bit	N/A	N/A

Notes:

1. Access type and reset value for all the reserved bits in the registers is read-only with value 0.
2. Register accesses should be word aligned and there is no support for a write strobe. WSTRB is not used internally.
3. Only the lower 7 bits (6:0) of the read and write address of the AXI4-Lite interface are decoded. This means that accessing address 0x00 and 0x80 results in reading the same address of 0x00.
4. Reads and writes to addresses outside this table do not return an error.

RST_CTRL Register

The Core Control register (0x00 offset) is described in [Table 2-3](#) and allows you to enable and disable the UHD-SDI RX IP core and apply a soft reset during core operation.

Table 2-3: RST_CTRL Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:10	Reserved	RO	0	Reserved
9	VID_IN_AXI4S_MOD_EN	R/W	0	Enable bit for Video-in-AXI4S core 1 – Video-in-AXI4S core is enabled 0 – Video-in-AXI4S core is disabled
8	SDIRX_BRIDGE_EN	R/W	0	Enable bit for SDI RX Bridge 1 – SDI RX Bridge is enabled 0 – SDI RX Bridge is disabled
7:4	Reserved	RO	0	Reserved
3	RST_EDH_ERRCNT	R/W	0	Clear rx_edh_errcnt register
2	RST_CRC_ERRCNT	R/W	0	Clear rx_crc_errcnt register
1	SRST	R/W	0	Soft reset for SDI RX IP core If 1 is written to this bit, all registers of SDI RX IP core will be resetted.
0	SDIRX_SS_EN	R/W	0	Enable bit for SDI RX IP core 1 – SDI RX IP core is enabled 0 – SDI RX IP core is disabled

MODULE_CTRL Register

The Module Control register (0x04 offset) is described in [Table 2-4](#) and allows you to control the UHD-SDI RX IP core and allows to change IP core functional modes.

Table 2-4: MODULE_CTRL Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:19	Reserved	RO	0	Reserved
18:16	RX_FORCED_MODE	R/W	0	RX forced mode 3'b000 : HD mode; 3'b001 : SD mode; 3'b010 : 3G mode; 3'b100 : 6G mode; 3'b101 : 12G mode with 11.88 Gb/s line rate; 3'b110 : 12G mode with 11.88/1.001 Gb/s line rate;
15:14	Reserved	RO	0	Reserved
13:8	RX_MODE_EN	R/W	0	RX mode enable Bit8: enable HD-SDI mode; Bit9: enable SD-SDI mode; Bit10: enable 3G-SDI mode; Bit11: enable 6G-SDI mode; Bit12: enable 12G-SDI 11.88 Gb/s mode; Bit13: enable 12G-SDI 11.88/1.001 Gb/s mode;
7:6	Reserved	RO	0	Reserved
5	RX_MODE_DET_EN	R/W	0	RX mode detection enable
4	RX_FRM_EN	R/W	0	RX frame enable
3:0	Reserved	RO	0	Reserved

Global Interrupt Enable Register (GLBL_IER)

Global interrupt enable register (0x0C offset) is described in [Table 2-5](#).

Table 2-5: GLBL_IER Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:1	Reserved	RO	0	Reserved
0	GLBL_INTRUPT_EN	R/W	0	Master enable for the device interrupt output to the system 1: Enabled—the corresponding Interrupt Enable register (IER) bits are used to generate interrupts 0: Disabled—Interrupt generation blocked irrespective of IER bits

Interrupt Status Register (ISR)

The Interrupt Status register (0x10 offset) is described in [Table 2-6](#) and captures the error and status information for the IP core.

Table 2-6: ISR bit mapping

Bits	Name	Access ⁽¹⁾	Default Value	Description
31:11	Reserved	RO	0	Reserved
10	UNDERFLOW_INTR	R/W1C	0	Video in to AXI-4 Stream core underflow indication
9	OVERFLOW_INTR	R/W1C	0	Video in to AXI-4 Stream core overflow indication
8:2	Reserved	RO	0	Reserved
1	VIDEO_UNLOCK_INTR	R/W1C	0	Asserted when incoming video pattern is unlocked
0	VIDEO_LOCK_INTR	R/W1C	0	Asserted when incoming video pattern is locked

Notes:

1. W1C = Write 1 to clear.

Interrupt Enable Register (IER)

The Interrupt Enable register (0x14 offset) is described in [Table 2-7](#) and allows you to selectively generate an interrupt at the output port for each error/status bit in the ISR. An IER bit set to 0 does not inhibit an error/status condition from being captured, but inhibits it from generating an interrupt.

Table 2-7: IER bit mapping

Bits	Name	Access	Default Value	Description
31:11	Reserved	RO	0	Set bits in this register to 1 to generate the required interrupts. Set to 0 to disable the interrupt. For a description of the specific interrupt you are enabling/disabling in this register see the ISR descriptions in Table 2-6 .
10	UNDERFLOW_INTR_EN	R/W	0	
9	OVERFLOW_INTR_EN	R/W	0	
8:2	Reserved	RO	0	
1	VIDEO_UNLOCK_INTR_EN	R/W	0	
0	VIDEO_LOCK_INTR_EN	R/W	0	

RX_ST352_VALID Register

RX_ST352_VALID register (0x18 offset) is described in [Table 2-8](#).

Table 2-8: RX_ST352_VALID Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:8	Reserved	RO	0	Reserved
7	RX_ST352_VLD_DS15	RO	0	Asserted high when ST352 data is valid on Data stream 15
6	RX_ST352_VLD_DS13	RO	0	Asserted high when ST352 data is valid on Data stream 13
5	RX_ST352_VLD_DS11	RO	0	Asserted high when ST352 data is valid on Data stream 11
4	RX_ST352_VLD_DS9	RO	0	Asserted high when ST352 data is valid on Data stream 9
3	RX_ST352_VLD_DS7	RO	0	Asserted high when ST352 data is valid on Data stream 7
2	RX_ST352_VLD_DS5	RO	0	Asserted high when ST352 data is valid on Data stream 5
1	RX_ST352_VLD_DS3	RO	0	Asserted high when ST352 data is valid on Data stream 3
0	RX_ST352_VLD_DS1	RO	0	Asserted high when ST352 data is valid on Data stream 1

RX_ST352_DATA_DS1 Register

RX_ST352_DATA_DS1 register (0x1C offset) is described in [Table 2-9](#).

Table 2-9: RX_ST352_DATA_DS1 Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:0	RX_ST352_DATA_DS1	RO	0	The ST 352 payload ID packet data bytes captured from data stream 1

RX_ST352_DATA_DS3 Register

RX_ST352_DATA_DS3 register (0x20 offset) is described in [Table 2-10](#).

Table 2-10: RX_ST352_DATA_DS3 Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:0	RX_ST352_DATA_DS3	RO	0	The ST 352 payload ID packet data bytes captured from data stream 3

RX_ST352_DATA_DS5 Register

RX_ST352_DATA_DS5 register (0x24 offset) is described in [Table 2-11](#).

Table 2-11: RX_ST352_DATA_DS5 Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:0	RX_ST352_DATA_DS5	RO	0	The ST 352 payload ID packet data bytes captured from data stream 5

RX_ST352_DATA_DS7 Register

RX_ST352_DATA_DS7 register (0x28 offset) is described in [Table 2-12](#).

Table 2-12: RX_ST352_DATA_DS7 Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:0	RX_ST352_DATA_DS7	RO	0	The ST 352 payload ID packet data bytes captured from data stream 7

RX_ST352_DATA_DS9 Register

RX_ST352_DATA_DS9 register (0x2C offset) is described in [Table 2-13](#).

Table 2-13: RX_ST352_DATA_DS9 Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:0	RX_ST352_DATA_DS9	RO	0	The ST 352 payload ID packet data bytes captured from data stream 9

RX_ST352_DATA_DS11 Register

RX_ST352_DATA_DS11 register (0x30 offset) is described in [Table 2-14](#).

Table 2-14: RX_ST352_DATA_DS11 Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:0	RX_ST352_DATA_DS11	RO	0	The ST 352 payload ID packet data bytes captured from data stream 11

RX_ST352_DATA_DS13 Register

RX_ST352_DATA_DS13 register (0x34 offset) is described in [Table 2-15](#).

Table 2-15: RX_ST352_DATA_DS13 Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:0	RX_ST352_DATA_DS13	RO	0	The ST 352 payload ID packet data bytes captured from data stream 13

RX_ST352_DATA_DS15 Register

RX_ST352_DATA_DS15 register (0x38 offset) is described in [Table 2-16](#).

Table 2-16: RX_ST352_DATA_DS15 Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:0	RX_ST352_DATA_DS15	RO	0	The ST 352 payload ID packet data bytes captured from data stream 15

VERSION Register

VERSION register (0x3C offset) is described in [Table 2-17](#).

Table 2-17: VERSION Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:0	VERSION	32-bit	32'h01_00_0_0_0 0	For uhd_sdi_rx_ss_v1_0, VERSION REGISTER will be 32'h01_00_0_0_00. <ul style="list-style-type: none"> • [31:24] - Core major version. • [23:16] - Core minor version. • [15:12] - Core version revision. • [11:8] - Core Patch details. • [7:0] - Internal revision.

SS_CONFIG Register

SS_CONFIG register (0x40 offset) is described in [Table 2-18](#).

Table 2-18: SS_CONFIG Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:2	Reserved	RO	0	Reserved
1	INC_RX_EDH_PROC	RO	1	This bit will be set if the IP core generated with INCLUDE_RX_EDH_PROCESSOR
0	Reserved	RO	0	Reserved

MODE_DET_STS Register

MODE_DET_STS register (0x44 offset) is described in [Table 2-19](#) and provides status about SDI mode detection.

Table 2-19: ODE_DET_STS Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:8	Reserved	RO	0	Reserved
7	RX_3G_LEVEL_B	RO	0	Asserted high when incoming stream is 3G-SDI level B
6:4	RX_ACT_STREAMS	RO	1	RX active data streams 3'b000 : 1 active stream; 3'b001 : 2 active streams (Default); 3'b010 : 4 active streams; 3'b011 : 8 active streams; 3'b100 : 16 active streams;

Table 2-19: ODE_DET_STS Register Bit Mapping (Cont'd)

Bits	Name	Access	Default Value	Description
3	RX_MODE_LOCKED	RO	1	RX mode locked indication If bit 5 of MODULE_CTRL is not set, then this bit will be set to 1'b1
2:0	RX_MODE	RO	0	3'b000 : HD-SDI Mode (default); 3'b001 : SD-SDI Mode; 3'b010 : 3G-SDI Mode; 3'b100 : 6G-SDI Mode; 3'b101 : 12G-SDI 11.88 Gb/s Mode; 3'b110 : 12G-SDI 11.88/1.001 Gb/s Mode;

TS_DET_STS Register

TS_DET_STS register (0x48 offset) is described in Table 2-20 and provides status about transport stream detection.

Table 2-20: TS_DET_STS Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:12	Reserved	RO	0	Reserved
11:8	RX_T_RATE	RO	0	This bit indicates the frame rate of the transport. This is not necessarily the same as the frame rate of the actual picture. This bit is only valid when RX_T_LOCKED is High. See Table 2-8 in the <i>SMPTE UHD-SDI Product Guide</i> (PG0205) [Ref 8] for details on the encoding of the bits.
7:4	RX_T_FAMILY	RO	4'hF	This bit indicates which family of video signals is being used as the transport of the SDI interface. This bit is only valid when RX_T_LOCKED is High. This bit does not necessarily identify the video format of the picture being transported. It only identifies the transport characteristics. See Table 2-7 in the <i>SMPTE UHD-SDI Product Guide</i> (PG0205) [Ref 8] for details on the encoding of the bits.
3:2	Reserved	RO	0	Reserved
1	RX_T_SCAN	RO	0	This bit indicates whether the transport is interlaced (Low) or progressive (High). This is not necessarily the same as the scan mode of the actual picture. This bit is only valid when RX_T_LOCKED is High
0	RX_T_LOCKED	RO	0	Asserted high when the transport detection function in the receiver has identified the transport format of the SDI signal.

RX_EDH_STS Register

RX_EDH_STS register (0x4C offset) is described in [Table 2-21](#).

Table 2-21: RX_EDH_STS Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:23	Reserved	RO	0	Reserved
22:19	RX_EDH_PKT_FLAGS	RO	0	This four error flags related to the most recently received EDH packet. See Table 2-5 in the <i>SMPTE UHD-SDI Product Guide</i> (PG0205) [Ref 8] for details on the encoding of the bits.
18:14	RX_EDH_ANC_FLAGS	RO	0	The ancillary error flag bits from the most recently received EDH packet. See Table 2-5 in the <i>SMPTE UHD-SDI Product Guide</i> (PG0205) [Ref 8] for details on the encoding of the bits.
13:9	RX_EDH_FF_FLAGS	RO	0	The full frame error flag bits from the most recently received EDH packet. See Table 2-5 in the <i>SMPTE UHD-SDI Product Guide</i> (PG0205) [Ref 8] for details on the encoding of the bits.
8:4	RX_EDH_AP_FLAGS	RO	0	The active picture error flag bits from the most recently received EDH packet. See Table 2-5 in the <i>SMPTE UHD-SDI Product Guide</i> (PG0205) [Ref 8] for details on the encoding of the bits.
3	Reserved	RO	0	Reserved
2	RX_EDH_ANC	RO	0	This output is asserted High when an ancillary data packet checksum error is detected.
1	RX_EDH_FF	RO	0	This bit is asserted High when the full field CRC calculated for the previous field does not match the FF CRC value in the EDH packet.
0	RX_EDH_AP	RO	0	This bit is asserted High when the active picture CRC calculated for the previous field does not match the AP CRC value in the EDH packet.

RX_EDH_ERRCNT_EN Register

RX_EDH_ERRCNT_EN register (0x50 offset) is described in [Table 2-22](#).

Table 2-22: RX_EDH_ERRCNT_EN Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:16	Reserved	RO	0	Reserved
15	EDH_PKT_CHKSUM_ERR	R/W	0	EDH packet checksum-error
14	AP_UES_ERR_EN	R/W	0	AP UES error

Table 2-22: RX_EDH_ERRCNT_EN Register Bit Mapping (Cont'd)

Bits	Name	Access	Default Value	Description
13	AP_IDA_ERR_EN	R/W	0	AP IDA error
12	AP_IDH_ERR_EN	R/W	0	AP IDH error
11	AP_EDA_ERR_EN	R/W	0	AP EDA error
10	AP_EDH_ERR_EN	R/W	0	AP EDH error
9	FF_UES_ERR_EN	R/W	0	FF UES error
8	FF_IDA_ERR_EN	R/W	0	FF IDA error
7	FF_IDH_ERR_EN	R/W	0	FF IDH error
6	FF_EDA_ERR_EN	R/W	0	FF EDA error
5	FF_EDH_ERR_EN	R/W	0	FF EDH error
4	ANC_UES_ERR_EN	R/W	0	ANC UES error
3	ANC_IDA_ERR_EN	R/W	0	ANC IDA error
2	ANC_IDH_ERR_EN	R/W	0	ANC IDH error
1	ANC_EDA_ERR_EN	R/W	0	ANC EDA error
0	ANC_EDH_ERR_EN	R/W	0	ANC EDH error

RX_EDH_ERRCNT Register

RX_EDH_ERRCNT register (0x54 offset) is described in [Table 2-23](#).

Table 2-23: RX_EDH_ERRCNT Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:16	Reserved	RO	0	Reserved
15:0	RX_EDH_ERRCNT	RO	0	SD-SDI mode EDH error counter. It increments once per field when any of the error conditions enabled by the RX_EDH_ERRCNT_EN register bit(s) occur during that field.

RX_CRC_ERR Register

RX_CRC_ERR register (0x58 offset) is described in [Table 2-24](#).

Table 2-24: RX_CRC_ERR Register Bit Mapping

Bits	Name	Access ⁽¹⁾	Default Value	Description
31:16	RX_CRC_ERR_CNT	RO	0	Cumulative CRC error count of Data stream 1 to 16
15	RX_CRC_ERR_DS16	R/W1C	0	CRC error indicator for each data stream 16
14	RX_CRC_ERR_DS15	R/W1C	0	CRC error indicator for each data stream 15
13	RX_CRC_ERR_DS14	R/W1C	0	CRC error indicator for each data stream 14
12	RX_CRC_ERR_DS13	R/W1C	0	CRC error indicator for each data stream 13
11	RX_CRC_ERR_DS12	R/W1C	0	CRC error indicator for each data stream 12. This bit remains asserted for one line time.
10	RX_CRC_ERR_DS11	R/W1C	0	CRC error indicator for each data stream 11
9	RX_CRC_ERR_DS10	R/W1C	0	CRC error indicator for each data stream 10
8	RX_CRC_ERR_DS9	R/W1C	0	CRC error indicator for each data stream 9
7	RX_CRC_ERR_DS8	R/W1C	0	CRC error indicator for each data stream 8
6	RX_CRC_ERR_DS7	R/W1C	0	CRC error indicator for each data stream 7
5	RX_CRC_ERR_DS6	R/W1C	0	CRC error indicator for each data stream 6
4	RX_CRC_ERR_DS5	R/W1C	0	CRC error indicator for each data stream 5.
3	RX_CRC_ERR_DS4	R/W1C	0	CRC error indicator for each data stream 4
2	RX_CRC_ERR_DS3	R/W1C	0	CRC error indicator for each data stream 3
1	RX_CRC_ERR_DS2	R/W1C	0	CRC error indicator for each data stream 2
0	RX_CRC_ERR_DS1	R/W1C	0	CRC error indicator for each data stream 1

Notes:

1. W1C = Write 1 to clear.

VIDEO_LOCK_WINDOW Register

VIDEO_LOCK_WINDOW register (0x5C offset) is described in [Table 2-25](#).

Table 2-25: VIDEO_LOCK_WINDOW Register Bit Mapping

Bits	Name	Access	Default Value	Description
31:16	Reserved	RO	0	Reserved
15:0	VIDEO_LOCK_WINDOW	R/W	0	Number of rx_clk cycles of stable video before asserting video is locked

Designing with the Core

This chapter includes guidelines and additional information to facilitate designing with the core.

General Design Guidelines

This section describes the steps required to turn a SMPTE UHD-SDI RX Subsystem into a fully functioning design with user-application logic.



IMPORTANT: *Not all implementations require all of the design steps listed here. Follow the logic design guidelines in this manual carefully.*

Use the Example Design as a Starting Point

Each instance of a SMPTE UHD-SDI RX Subsystem that is created is delivered with an example design that can be implemented in Xilinx FPGA. This design can be used as a starting point for your own design or can be used to troubleshoot the user application, if necessary.

Know the Degree of Difficulty

SMPTE UHD-SDI RX Subsystem design is challenging to implement in any technology, and the degree of difficulty is further influenced by:

- Maximum system clock frequency
- Targeted device architecture
- Nature of the user application

All SMPTE UHD-SDI RX Subsystem implementations require careful attention to system performance requirements. Pipelining, logic mappings, placement constraints and logic duplications are all methods that help boost system performance.

Keep It Registered

To simplify timing and increase system performance in an FPGA design, keep all inputs and outputs registered with flip-flops between the user application and the subsystem. Registering signals might not be possible for all paths, but doing so simplifies timing analysis and makes it easier for the Xilinx tools to place-and-route the design.

Recognize Timing Critical Signals

The XDC file provided with the example design for the core identifies the critical signals and the timing constraints that should be applied.

Make Only Allowed Modifications

The SMPTE UHD-SDI RX Subsystem is not user modifiable. Any modifications might have adverse effects on the system timings and protocol compliance. Supported user configurations of the SMPTE UHD-SDI RX Subsystem can only be made by selecting options from the Vivado® Integrated Design Environment (IDE).

Clock Frequency Selection

SMPTE UHD-SDI RX Subsystem inherently has multiple clock domains and has many CDC paths across the core. It is recommended to use maximum allowed clock frequency to reduce the uncertainty due to cdc paths.

Clocking

The subsystem clocks are described in [Table 3-1](#). Clock frequencies should be selected to match the throughput requirement and SDI standard.

Table 3-1: Subsystem Clocks

Clock Name	Description
s_axi_aclk	AXI4-Lite clock used by the register interface of all IP cores in the subsystem. Frequency range could be 50 MHz to 150 MHz
sdi_rx_clk	Core clock for UHD-SDI RX core. Refer Table 3-2 for more details.

Table 3-1: Subsystem Clocks (Cont'd)

Clock Name	Description
video_out_clk	Clock used for Video data conversion from SDI data stream. In order to support 12G-SDI for 10-bit YUV 4:2:2 in 2 PPC ⁽¹⁾ , clock must set to maximum 300MHz. $2*(BPC)^{(2)}*(PPC)*clock = 2*10*2*300MHz = 12 Gbps$ video_out_clk for SMPTE UHD-SDI RX Subsystem must not be less than sdi_rx_clk (will cause underflow). User need to cautious on overflow if using value too much higher than sdi_rx_clk.

Notes:

1. BPC is Bits per component that is set to 10 since Subsystem supports 10-bit YUV4:2:2.
2. PPC is Pixel Per Clock that is set to 2 by SDI bridge.

The frequency of sdi_rx_clk of UHD-SDI RX core is given [Table 3-2](#).

Table 3-2: UHD-SDI RX Clock

SMPTE Standard	Supported Data Stream	Clock Frequency (in MHz)
SD-SDI	1	148.5 (27 MHz sampling at rx_sd_ce with 5-6-5-6 cadence)
HD-SDI	2	74.25
3G-SDI Level A	2	148.5
3G-SDI Level B	4	148.5
6G-SDI	8	148.5
12G-SDI	8	297

More clocking details can be referenced in the *Clocking* section the *SMPTE UHD-SDI Product Guide* [\[Ref 8\]](#).

Resets

The subsystem has three reset ports:

- **s_axi_arstn**: Active-Low reset for the AXI4-Lite register interface and synchronous with s_axi_aclk.
- **video_out_arstn**: Active-Low reset for the subsystem blocks and synchronous with video_out_clk.
- **sdi_rx_rst**: Active-High reset for the UHD-SDI RX core and synchronous with sdi_rx_clk. Refer to the *Clocking* section the *SMPTE UHD-SDI Product Guide* [\[Ref 8\]](#).

Table 3-3 summarizes all resets available to the SMPTE UHD-SDI RX Subsystem and the components affected by them.

Table 3-3: Core Resets

Sub-Core	s_axi_arstn	video_out_arstn	sdi_rx_rst
SMPTE UHD-SDI RX	Connected to s_axi_aresetn core port	Connected to axis_rstn core port	Connected to rx_rst core port
SDI RX to Video Bridge	NA	NA	Connected to rst core port
Video In to AXI-4 Stream	NA	NA	NA

Note: The effect of each reset (s_axi_arstn, video_out_arstn, sdi_rx_rst) is determined by the ports of the sub-cores to which they are connected. See the individual sub-core product guides for the effect of each reset signal.

Design Flow Steps

This chapter describes customizing and generating the core, constraining the core, and the simulation, synthesis and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows in the IP Integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [\[Ref 1\]](#)
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [\[Ref 2\]](#)
- *Vivado Design Suite User Guide: Getting Started* (UG910) [\[Ref 3\]](#)
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [\[Ref 4\]](#)

Customizing and Generating the Core

This section includes information about using Xilinx tools to customize and generate the core in the Vivado® Design Suite.

Vivado Integrated Design Environment

You can customize the IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [\[Ref 2\]](#) and the *Vivado Design Suite User Guide: Getting Started* (UG910) [\[Ref 3\]](#).

Note: Figures in this chapter are illustrations of the Vivado IDE. The layout depicted here might vary from the current version.

Core Configuration Tab

Figure 4-1 shows the Core Configuration tab for customizing the UHD-SDI RX Subsystem.

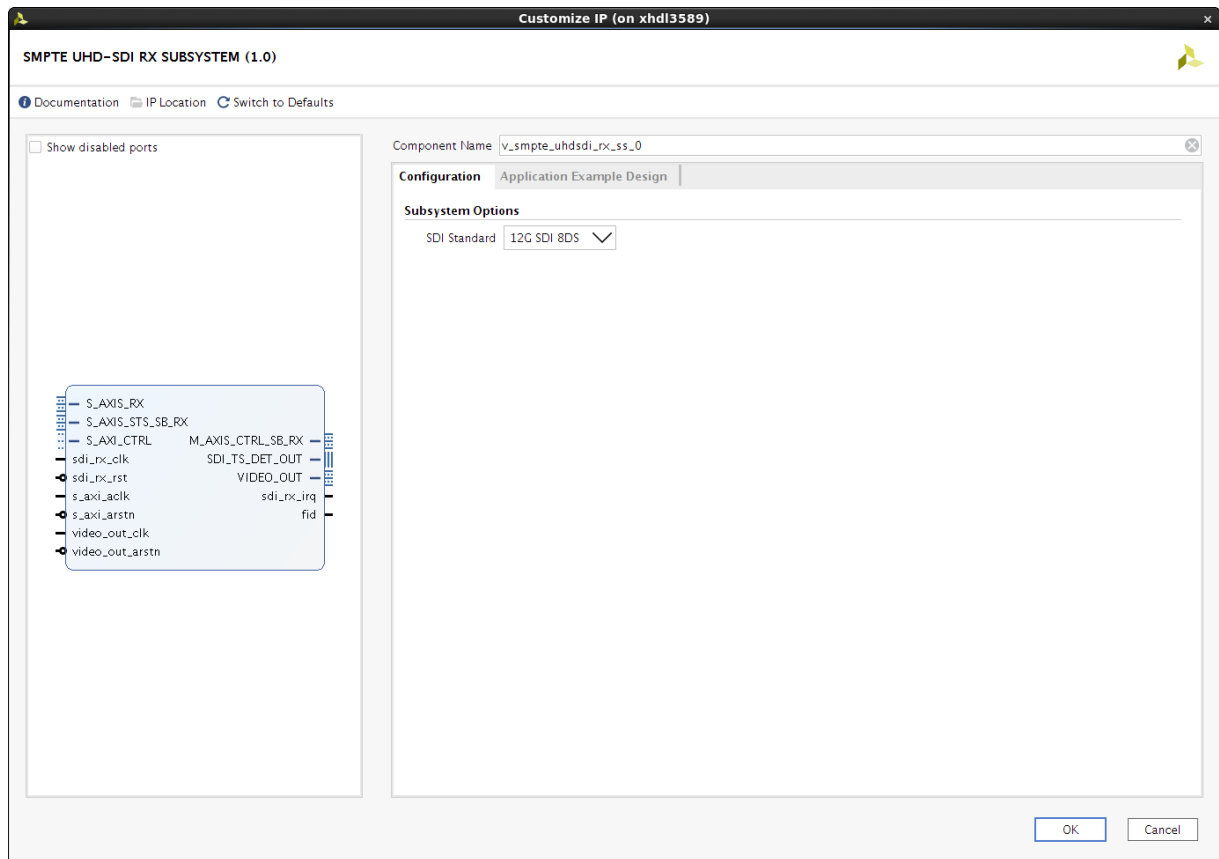


Figure 4-1: Subsystem Configuration Tab

Component Name: The Component Name is the base name of the output files generated for this core.



IMPORTANT: The name must begin with a letter and be composed of the following characters: a to z, A to Z, 0 to 9 and "_."

Core Parameters

- **SDI Standard:** Specifies the SDI standard. Available options:
 - 3G SDI
 - 6G SDI
 - 12G SDI 8DS

Application Example Design Tab

Figure 4-2 shows the Application Example Design tab for using the UHD-SDI RX example design. Refer to Chapter 5, Example Design for more information on the Example Design.

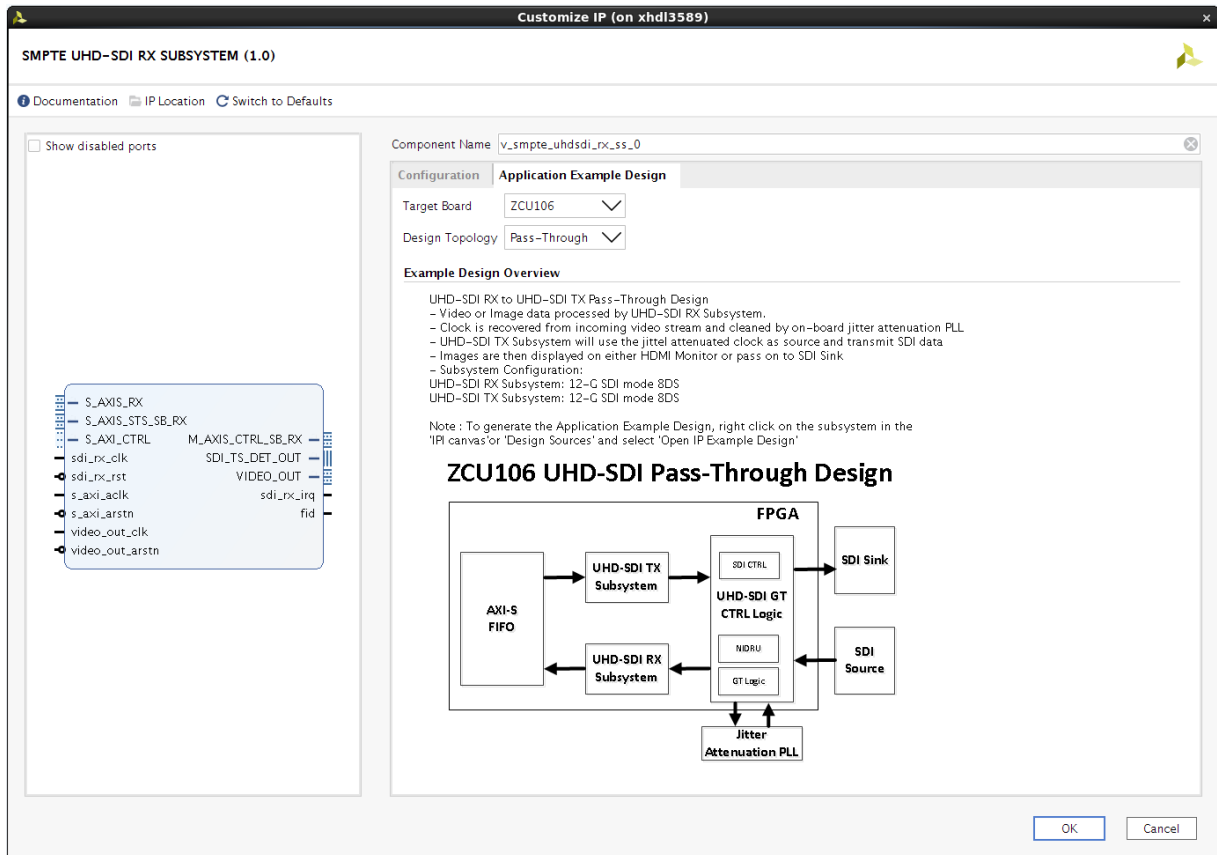


Figure 4-2: ZCU102 Pass-Through Application Example Design

Target Board: Target board on which the Application example design to be built. Supported value is ZCU106.

Design Topology: Type of configuration for Application Example Design. Available option is Pass-Through.

User Parameters

Table 4-1 shows the relationship between the fields in the Vivado IDE and the User Parameters (which can be viewed in the Tcl Console).

Table 4-1: Vivado IDE Parameter to User Parameter Relationship

Vivado IDE Parameter to User Parameter Relationship		
Vivado IDE Parameter/Value ⁽¹⁾	User Parameter/Value	Default Value
Core Parameters		
SDI Standard	C_LINE_RATE	12G SDI 8DS
Parameter values are listed in the table where the Vivado IDE parameter value differs from the user parameter value. Such values are shown in this table as indented below the associated parameter.		

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Constraining the Core

This section contains information about constraining the core in the Vivado Design Suite.

Required Constraints

This section defines the additional constraint requirements for the subsystem. Constraints are provided with a Xilinx Design Constraints (XDC) file. An XDC is provided with the HDL example design to give a starting point for constraints for your design.

Device, Package, and Speed Grade Selections

This section is not applicable for this subsystem.

Clock Frequencies

See [Clocking in Chapter 3](#).

Clock Management

The SMPTE UHD-SDI RX Subsystem generates the required clock constraints when generated using out-of-context mode with <component_name>_ooc.xdc. You can use these or update as required for other clock constraints.

Clock Placement

This section is not applicable for this subsystem.

Banking

This section is not applicable for this subsystem.

Transceiver Placement

This section is not applicable for this subsystem.

I/O Standard and Placement

This section is not applicable for this subsystem.

Simulation

This section contains information about simulating IP in the Vivado Design Suite. For comprehensive information about Vivado simulation components, as well as information about using supported third-party tools, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 4].

Synthesis and Implementation

This section contains information about synthesis and implementation in the Vivado Design Suite. For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Example Design

This chapter contains information about the example design provided in the Vivado® Design Suite.

ZCU106 UHD-SDI Pass-Through Example Design

Introduction

The UHD-SDI Pass-through example design, shown in [Figure 5-1](#), is built using UHD-SDI TX and RX Subsystems. Video or image data is received and processed by UHD-SDI RX Subsystem. The clock is recovered by Xilinx® Ultrascale™+ GTH transceiver (RX) and feed to on-board PLL for jitter attenuation. A jitter-attenuated clock is used as a reference clock by the GTH transceiver for the TX data path. An AXI4-Stream FIFO is used for synchronization and temporal storage between the UHD-SDI RX Subsystem and UHD-SDI TX Subsystem. The UHD-SDI TX Subsystem transmits SDI data from AXI4-Stream FIFO after the application programs the UHD-SDI TX Subsystem subcore registers based on the received SDI stream and ST-352 payload packet data. The example design application software runs on the Zynq® UltraScale+ MPSoC ARM processor subsystem (PS) and is fully software controlled.

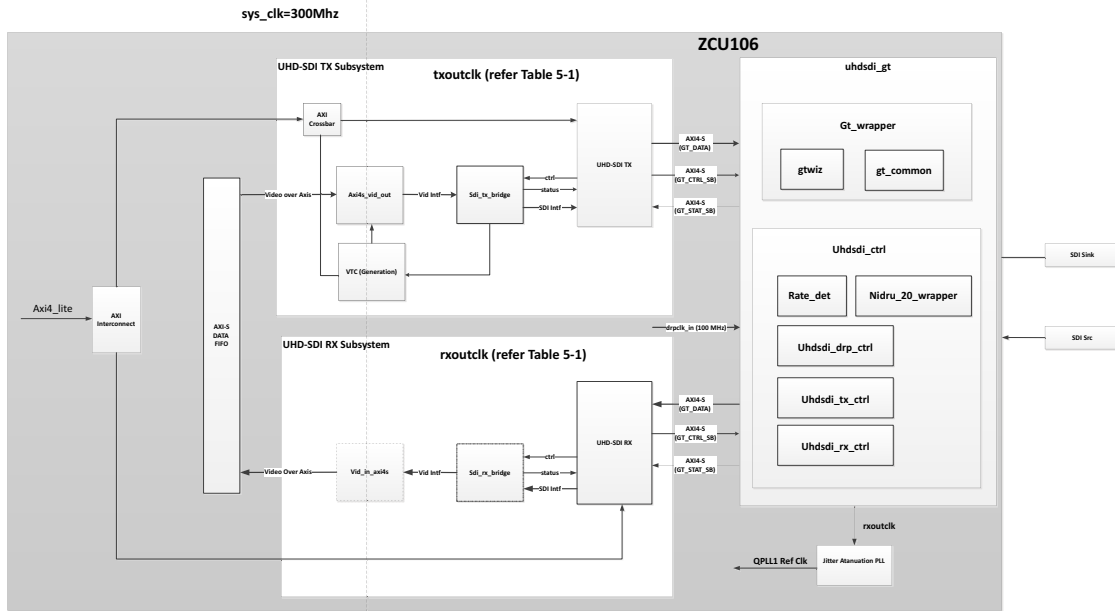


Figure 5-1: ZCU106 UHD-SDI Pass-Through Design

Note: Zynq UltraScale+ MPSoC PS is not shown in the Figure 5-1 for simplicity.

Using UltraScale+ GTH Transceivers for SDI Interfaces

The information in this section is intended to supplement, not replace, the information in the *UltraScale Architecture GTH Transceivers User Guide* (UG576) [Ref 11]. This information highlights features and operating requirements of the GTH transceivers that are of particular importance for UHD-SDI applications.

In the example design, the naming convention used in the *UltraScale Architecture GTH Transceivers User Guide* (UG576) [Ref 11] for the GTH transceiver ports is followed. This convention is to use only the base name of a port. When the UltraScale FPGAs Transceiver Wizard is used to create a GTH wizard module, all input ports names have a suffix of `_in` and all outputs have a suffix of `_out`. For example, when a port named `txp11clkssel` is mentioned in this document, the actual name of that port in the GTH wrapper would be `txp11clkssel_in`.

There are several clocks required in applications using GTH transceivers. The SDI protocol, which does not allow for clock correction by stuffing and removing extra data in the data stream, requires careful attention to how these clocks are generated and used in the application. GTH transceivers require reference clocks to operate. The reference clocks are used by phase-locked loops (PLLs) in the GTH transceiver quad to generate serial clocks for the receiver and transmitter sections of each transceiver. As described in [GTH Transceiver Reference Clocks](#), the serial bit rate of the GTH transmitter is an integer multiple of the reference clock frequency it is using. Furthermore, the data rate of the video provided to the input of the SDI transmitter datapath must also exactly match (or be a specific multiple of)

the frequency of the reference clock used by the GTH transmitter. Consequently, you must determine how to generate the transmitter reference clock so that it is frequency-locked exactly with the data rate of the video stream being transmitted.

The GTH transmitter clocking is handled by the Transmitter User Clocking Network Helper Block when enabled during GT IP generation from UltraScale FPGAs Transceiver Wizard. The `txusrclk` and `txusrclk2` output is driven by a BUFG_GT within the helper block and its frequency is exactly equal to the word rate of the data that must enter the `txdata` port of the GTH transmitter. The `txusrclk` and `txusrclk2` are generated in the GTH transmitter by dividing the serial clock from the PLL down to the word rate. Refer to the *UltraScale FPGAs Transceivers Wizard* (PG182) [Ref 12] for more details on Transmitter User Clocking Network Helper Block.

The GTH receiver reference clock, however, does not need an exact relationship with the bit rate of the incoming SDI signals. This is because the clock and data recovery (CDR) unit in the GTH receiver can receive bit rates that are up to $\pm 1,250$ ppm (≤ 6.6 Gbps) or ± 200 ppm (> 8.0 Gbps) away from the nominal bit rate as set by the reference clock frequency. This allows the receiver reference clock to be generated by local oscillators that have no exact frequency relationship to the incoming SDI signal. The GTH receiver generates a recovered clock that is frequency-locked to the incoming SDI bit rate. These clocks are output as `rxusrclk` and `rxusrclk2` ports of the Receiver User Clocking Network Helper Block from the GTH Wizard IP and are driven by BUFG_GT. As is described in more detail later in this application note, `rxusrclk` and `rxusrclk2` are true recovered clocks when receiving all SDI line rates except when receiving SD-SDI signals.

One additional clock is required for SDI applications. This is a free-running, fixed-frequency clock that is used as the clock for the dynamic reconfiguration port (DRP) of the GTH transceiver. This same clock is also usually supplied to the control module in the SDI wrapper where it is used for timing purposes. The valid frequency range for this clock is stated in the UltraScale FPGAs Transceiver Wizard and normally is ranging from 3.125 to 200 MHz. The frequency of this clock does not require any specific relationship relative to other clocks or data rates of the SDI application. This clock must not change frequencies when the SDI mode changes. It must remain running at the same nominal frequency at all times. It also must never stop while the SDI application is active. This clock can be used for all SDI interfaces in the device.

The frequency of the `rxusrclk` and `txusrclk` depend on the SDI mode and the width of the GTH transceiver's `rxdata` and `txdata` ports. This relationship is fixed by the architecture of the GTH transceiver. The RX and the TX both use clock enables to throttle the data stream transfer data rate because, in some cases, the data rate on the data streams is less than the frequency of the clock. Table 5-1 shows the relationships between SDI mode, number of active data streams, `rxdata`/`txdata` port widths, `rxoutclk`/`txoutclk` frequencies, and clock enable cadences. The clock enable cadences are given in number of clocks between assertions of the clock enable over two data word cycles where 1/1 means that the clock enable is asserted every clock cycle, 2/2 indicates assertion every other clock cycle (50% duty cycle), 4/4 indicates assertion every fourth clock cycle (25% duty cycle), and 5/6 indicates that the clock enable alternates between assertion every 5 or 6 clock cycles, to

average once every 5.5 clock cycles (one instance of 5 clock cycles between High pulses on the clock enable followed by one instance of six clock cycles between High pulses on the clock enable, with this pattern repeating).

Table 5-1: Clock Frequencies and Clock Enable Requirements

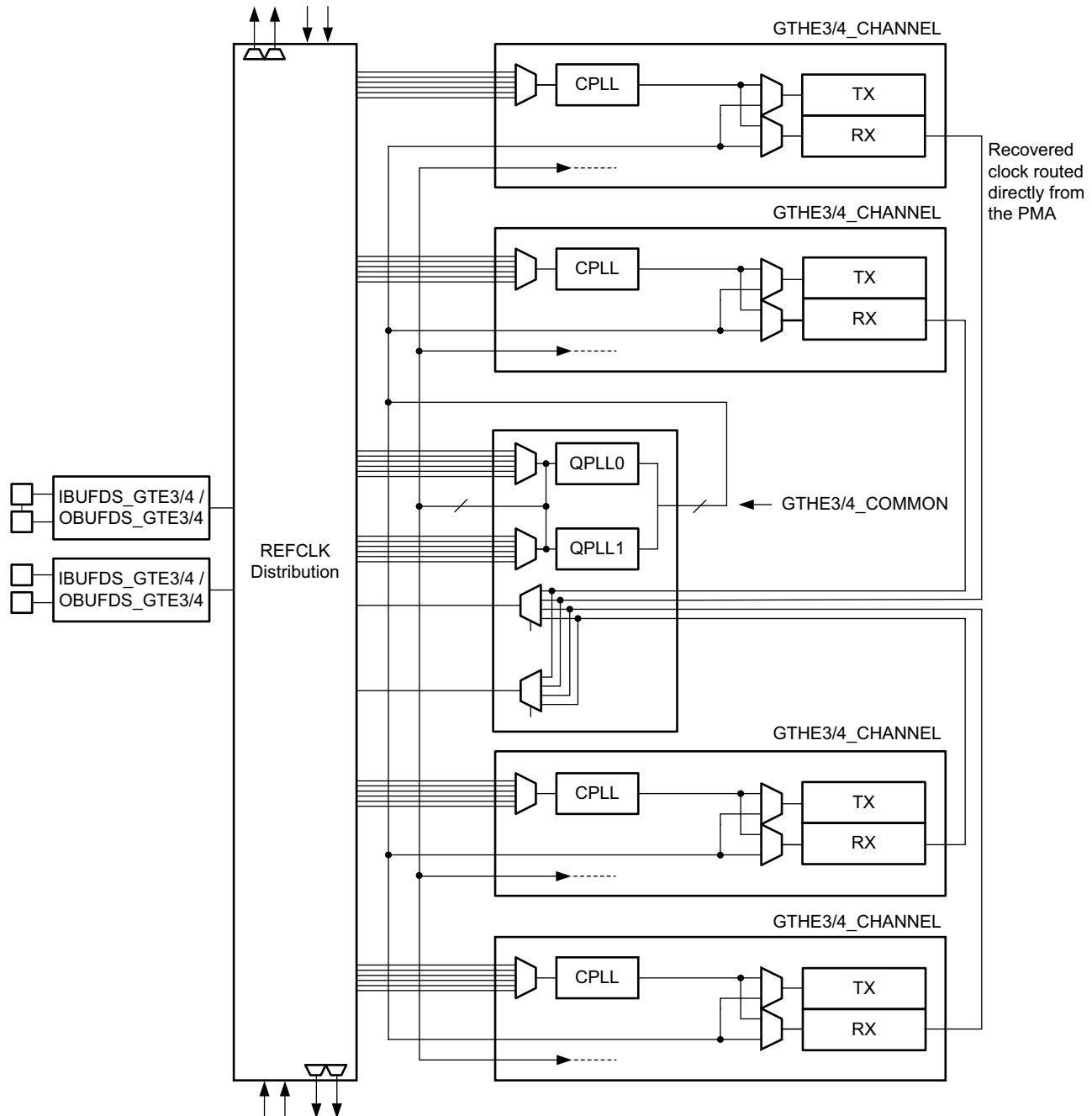
SDI-Mode	Active Data Streams	RX/TXDATA Bit Width	RX/TXOUTCLK Frequency	Clock enable
SD-SDI	1	20	148.5 MHz	5/6
HD-SDI	2	20	74.25 or 74.25/1.001 MHz	1/1
3G-SDI A	2	20	148.5 or 148.5/1.001 MHz	1/1
3G-SDI B	4	20	148.5 or 148.5/1.001 MHz	2/2
6G-SDI	4	40	148.5 or 148.5/1.001 MHz	1/1
6G-SDI	8	40	148.5 or 148.5/1.001 MHz	2/2
12G-SDI	8	40	297 or 297/1.001 MHz	2/2
12G-SDI	16	40	297 or 297/1.001 MHz	4/4

GTH Transceiver Reference Clocks

UltraScale+ GTH transceivers are grouped into quads. Each quad contains four GTHE4_CHANNEL transceiver primitives and one GTHE4_COMMON primitive containing two quad PLLs (QPLL0 and QPLL1) as shown in [Figure 5-2](#). The clock generated by the QPLL0 and QPLL1 are distributed to all four transceivers in the quad. Each GTHE4_CHANNEL has its own PLL called the Channel PLL (CPLL), which can provide a clock to the RX and TX of that transceiver only. Each RX and TX unit in the quad can be individually configured to use either/both QPLL0 or/and QPLL1 or the CPLL as its clock source. Furthermore, any RX or TX unit can dynamically switch its clock source between QPLL0, QPLL1 and CPLL. This configuration and the dynamic switching capability are particularly useful for SDI applications.



IMPORTANT: *The CPLL and QPLL have maximum line rates of 6.25 Gbps and 16.375 Gbps, respectively. This means that CPLL can only be used up to 6G-SDI line rate while QPLLs can support up to 12G-SDI. It is important to note that for -1 speed grade UltraScale+ GTH transceivers, the CPLL has a maximum line rate of 4.25 Gbps and therefore can only support up to 3G-SDI. Again, this is a limitation only of -1 speed grade devices. See GTH Transceiver Switching Characteristics section of Kintex UltraScale+ Architecture Data Sheet: DC and AC Switching Characteristics (DS922) [Ref 13] for details.*



UG576_ct_01_042817

Figure 5-2: UltraScale+ GTH Transceiver Quad Configuration

Typical UHD-SDI applications require the GTH transceivers to support nine different bit rates:

- 270 Mb/s for SD-SDI
- 1.485 Gb/s for HD-SDI
- 1.485/1.001 Gb/s for HD-SDI

- 2.97 Gb/s for 3G-SDI
- 2.97/1.001 Gb/s for 3G-SDI
- 5.94 Gb/s for 6G-SDI
- 5.94/1.001 Gb/s for 6G-SDI
- 11.88 Gb/s for 12G-SDI
- 11.88/1.001 Gb/s for 12G-SDI

The CDR unit in the RX section of the GTH transceiver can support receiving bit rates that are up to +/-1250 ppm from the reference frequency at bit rates less than 6.6 Gb/s. HD-SDI, 3G-SDI, 6G-SDI, and 12G-SDI each have two bit rates that differ by exactly 1000 ppm. For HD-SDI, 3G-SDI, and 6G-SDI, both bit rates can be received using a single reference clock frequency. That same reference clock frequency can also support reception of SD-SDI. Thus, for all SDI modes except 12G-SDI, just a single RX reference clock frequency is required. However, at 12G-SDI rates, the CDR unit has only ± 200 ppm tolerance relative to the reference clock frequency. Thus two different reference clock frequencies are needed to receive the two 12G-SDI bit rates. These two reference clock frequencies are typically 148.5 MHz to receive 11.88 Gb/s and 148.5/1.001 MHz to receive 11.88/1.001 Gb/s.

Therefore, most SDI applications provide two separate reference clocks to the GTH quad. Usually, the supplied reference frequency pair are 148.5 MHz and 148.5/1.001 MHz. This application note always refers to the reference clock frequency pair 148.5 MHz and 148.5/1.001 MHz.

The source of the GTH transceiver reference clocks is very application specific. The receiver reference clock source can be a local oscillator because it does not need to match the incoming SDI bit rate exactly. However, because the GTH transmitter line rate is always an integer multiple of the reference clock frequency, the frequency of the transmitter reference clock must be exactly related to the data rate of the transmitted data. Most often, the transmitter reference clocks are generated by genlock PLLs, thereby deriving the GTH transmitter line rate from the studio video reference signal. In some cases, such as the SDI pass-through connection, the transmitter line rate is derived from the recovered clock of the GTH receiver that is receiving the SDI signal. In such cases, an external PLL is required to reduce the jitter on the recovered clock before using it as the transmitter reference clock.

In a typical UHD-SDI application, the two reference clocks are connected to the QPLL0 and QPLL1. The RX and TX units of each transceiver in the quad dynamically switch between the PLL clocks, depending on the bit rate that is required at the moment. The GTH `txsysclkssel` and `rxsysclkssel` ports are used to select the TX and RX units serial clock source between the PLLs. This common configuration for SDI applications is shown in [Figure 5-3](#). In this [Figure 5-3](#), multiplexers that are not used dynamically in the implementation have been replaced with wires and the reference clock routing between quads is not shown. It is also possible to connect one reference clock to CPLL and the other to QPLL0/1 provided that only one 12G-SDI bit rate is supported.

Also, each GTH RX and TX unit has a serial clock divider that divides the selected clock by several selectable integer powers of two. This allows, for example, all of the RX units in the quad to use the same clock frequency from the QPLL but operate at different lines rates by using different serial clock divider values. This is very useful for SDI interfaces because the 3G-SDI, 6G-SDI and 12G-SDI bit rates are exactly twice as fast the HD-SDI, 3G-SDI and 6G-SDI bit rates respectively. And, for 270 Mb/s SD-SDI, the GTH transceiver runs at the 3G-SDI line rate using 11X oversampling techniques. The ability of the RX and TX units to locally divide the clock source by four divisors that differ by a factor of two is important, allowing reception and transmission of all SDI bit rates using just two reference clock frequencies.

The serial clock divider value of each RX and TX unit can be changed dynamically through the DRP, by using the RXOUT_DIV and TXOUT_DIV attributes.

The configuration shown in [Figure 5-3](#) is an optimal solution for most SDI applications for several reasons:

- The receivers can receive all SDI bit rates when using QPLL0 and QPLL1 to provide the serial clock derived from that reference clocks to all receivers in the quad.
- The transmitters have the flexibility to dynamically switch between the clocks from QPLL0 and QPLL1 to get both frequencies they need to transmit all supported SDI bit rates.
- All four receivers and all four transmitters in the quad are fully independent and can each be running at different SDI bit rates and can dynamically switch between bit rates without disrupting the other RX or TX units.
- For genlocked applications, modern genlock PLLs usually can simultaneously provide both required reference clock frequencies from the synchronization reference input signal.

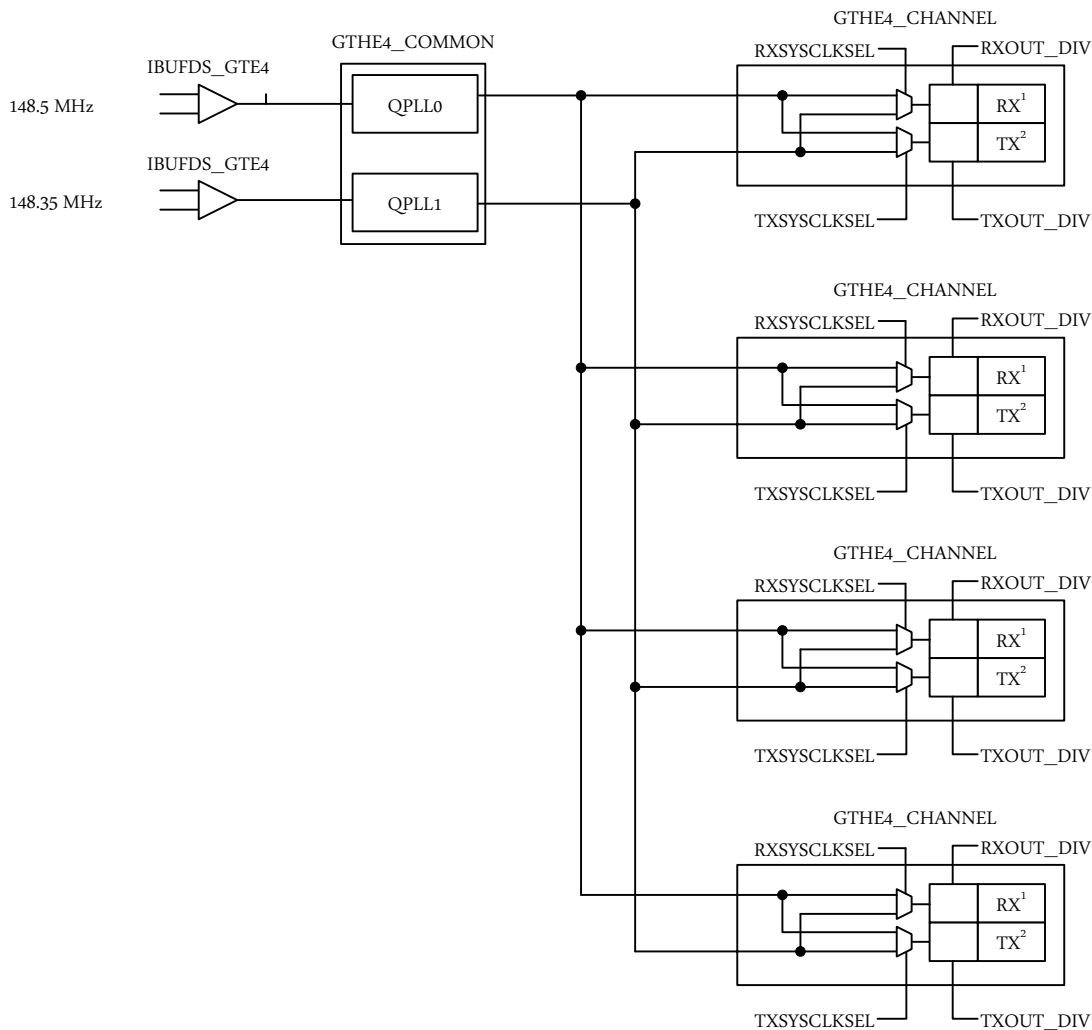


Figure 5-3: Typical GTH Reference Clock Implementation for SDI

Notes:

1. GTH RX interface and internal bit width are dynamically changed through RX_DATA_WIDTH and RX_INT_DATAWIDTH DRP attributes depending on the current SDI Mode and data stream inter-leaving pattern.
2. GTH TX interface and internal bit width are dynamically changed through TX_DATA_WIDTH and TX_INT_DATAWIDTH DRP attributes depending on the current SDI Mode and data stream inter-leaving pattern.

In some SDI applications, it might be necessary for different SDI transmitter to be running at slightly different bit rates even though they are transmitting at the same nominal bit rate. This is often the case with SDI routers where the bit rate of each TX must exactly match the bit rate of the SDI signal received by the SDI RX to which the TX is currently connected. In these cases, two transmitters that are transmitting at the same nominal bit rate, in fact, have bit rates that differ by a few ppm. Supporting such applications is possible with the UltraScale+ GTH quad architecture because each TX unit has exclusive use of its own CPLL. But to accomplish this, each CPLL must be provided with its own individual reference clock frequency, and the number of GTH reference clock inputs is limited. There are two reference clock inputs per GTH quad. A quad can use reference clocks from the quad above and the

quad below. Thus, it is possible to provide some GTH quads in the device with five different reference clock frequencies (one for the RX and four for the four TX units), but overall, there are obviously not enough reference clock inputs to allow every GTH TX in the device to have its own reference clock. The PICXO technique can be very useful in these cases because it allows a GTH TX to be pulled by a few hundred ppm away from the frequency of its serial clock. Thus, applications where the bit rate of each SDI TX must be individually locked to the bit rate of the received SDI signal can be implemented by using common reference clocks as in Figure 5-3 and then using the PICXO technique with each GTH TX to set the exact bit rate of each SDI transmitter individually. This application note does not cover the PICXO technique. For further information about using PICXO, contact Xilinx technical support.

Clocking

QPLL0 is allocated for UHD-SDI RX Transceiver and QPLL1 for UHD-SDI TX in this pass-through design. The reference clock for QPLL1 comes from si5324 chip output. Thus, QPLL1 reference clock connection is fixed. QPLL0 reference clock is fixed to 148.5 MHz which comes from on-board si570 chip. Figure 5-4 shows the clocking used in the UHD-SDI example design.

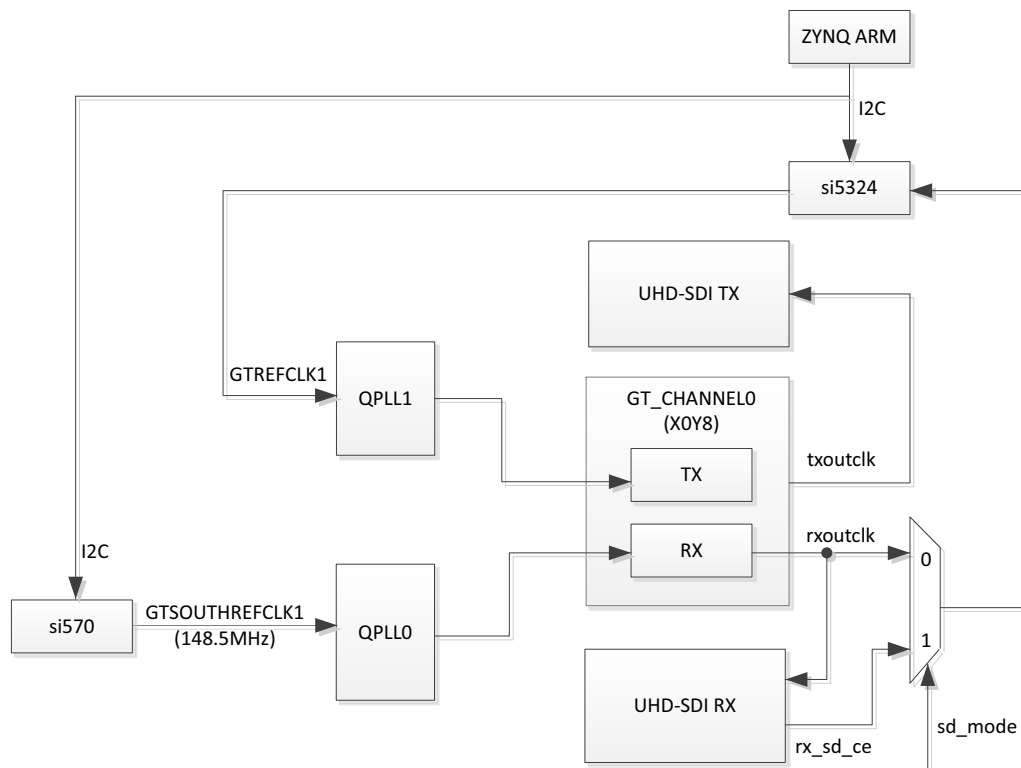


Figure 5-4: ZCU106 UHD-SDI Pass-Through Example Design Clocking

Table 5-2 shows the clock frequency at different part of the system for different SDI mode:

Table 5-2: HD-SDI Example Design Clock Frequency Ranges

SDI Mode	Tx_m/ Rx_m	QPLL0 Ref clk (MHz)	QPLL1 Ref Clk (MHz)	txoutclk (MHz)	si5328 Input (MHz)	si5328 output (MHz)	txoutclk (MHz)
SD-SDI	N.A.	148.5	148.5	148.5 rx_sd_ce =27	rx_sd_ce =27	148.5	148.5
HD-SDI	0	148.5	148.5	74.25	74.25	148.5	148.5
HD-SDI	1	148.5	148.5 /1.001	74.25 /1.001	74.25 /1.001	148.5 /1.001	148.5 /1.001
3G-SDI/6G-SDI	0	148.5	148.5	148.5	148.5	148.5	148.5
3G-SDI/6G-SDI	1	148.5	148.5 /1.001	148.5 /1.001	148.5 /1.001	148.5 /1.001	148.5 /1.001
12G-SDI	0	148.5	148.5	297	297	148.5	297
12G-SDI	1	148.5	148.5 /1.001	297 /1.001	297 /1.001	148.5 /1.001	297 /1.001

Note: For 6G-SDI and 12G-SDI, 8 native SDI Data Streams (DS) is assumed.

For GT TX and RX data path, the reference clock requirement for data paths are different. For GT TX, for integer and fractional frame rate, PLL reference clock must be different frequency, clock/1.000 for integer frame rate and clock/1.001 for fractional frame rate. For RX data path PLL reference clock can be same for integer and fractional frame rate.

Transceiver Configuration in Example Design

The UHD-SDI example design uses `uhdsdi_gt_v1_0` core to configure Ultrascale+ GTH transceivers and provide options to select the Transceiver reference clocking. The core also generates control modules that are required to program the transceiver using DRP interface and NI-DRU modules for RX SD-SDI mode. Below Figure 5-x provides GUI configuration that is used in the ZCU106 UHD-SDI Pass-Through application example design.

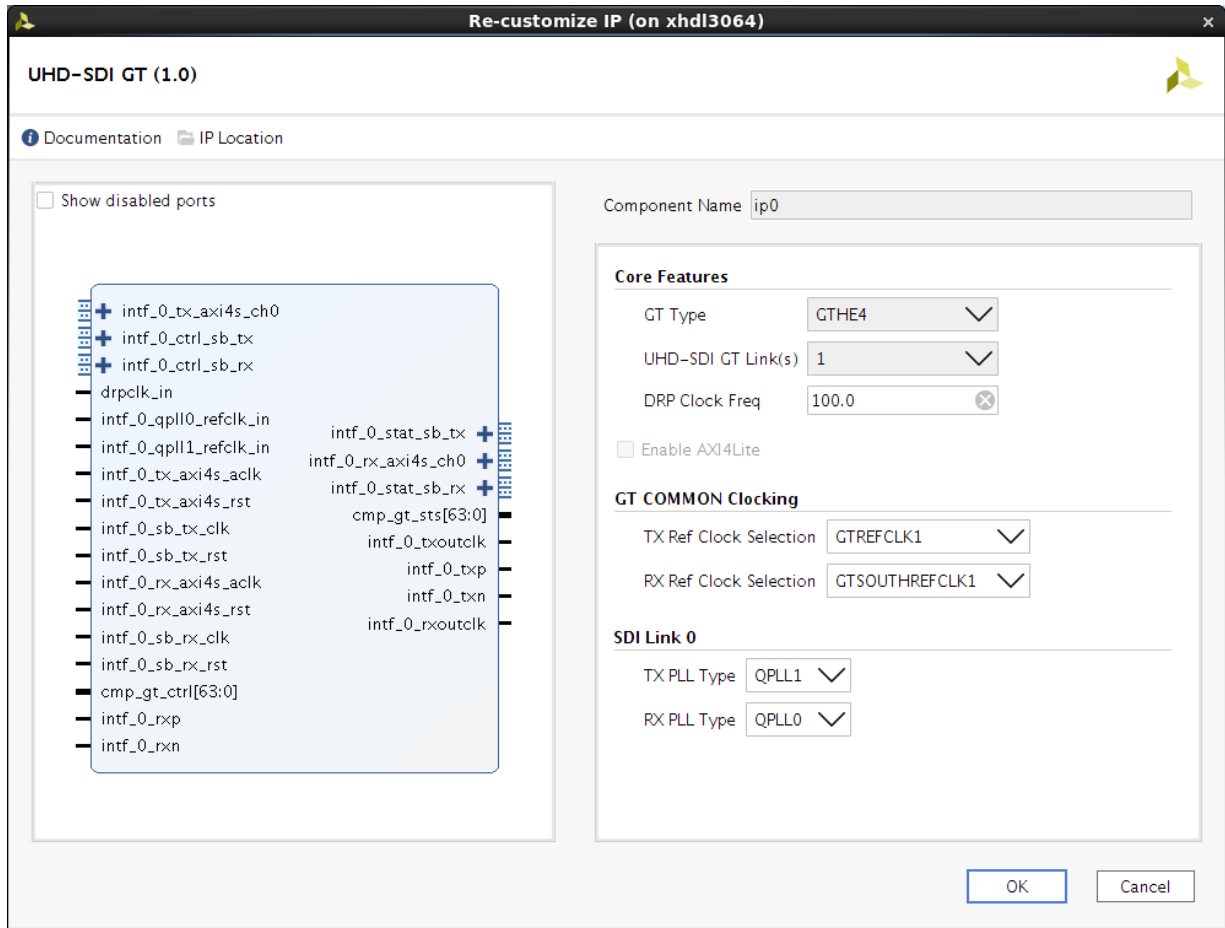


Figure 5-5: uhdsdi_gt Core Configuration

Core Parameters

This section provides details on the uhdsdi_gt_v1_0 core options to configure the Transceiver based on user requirement:

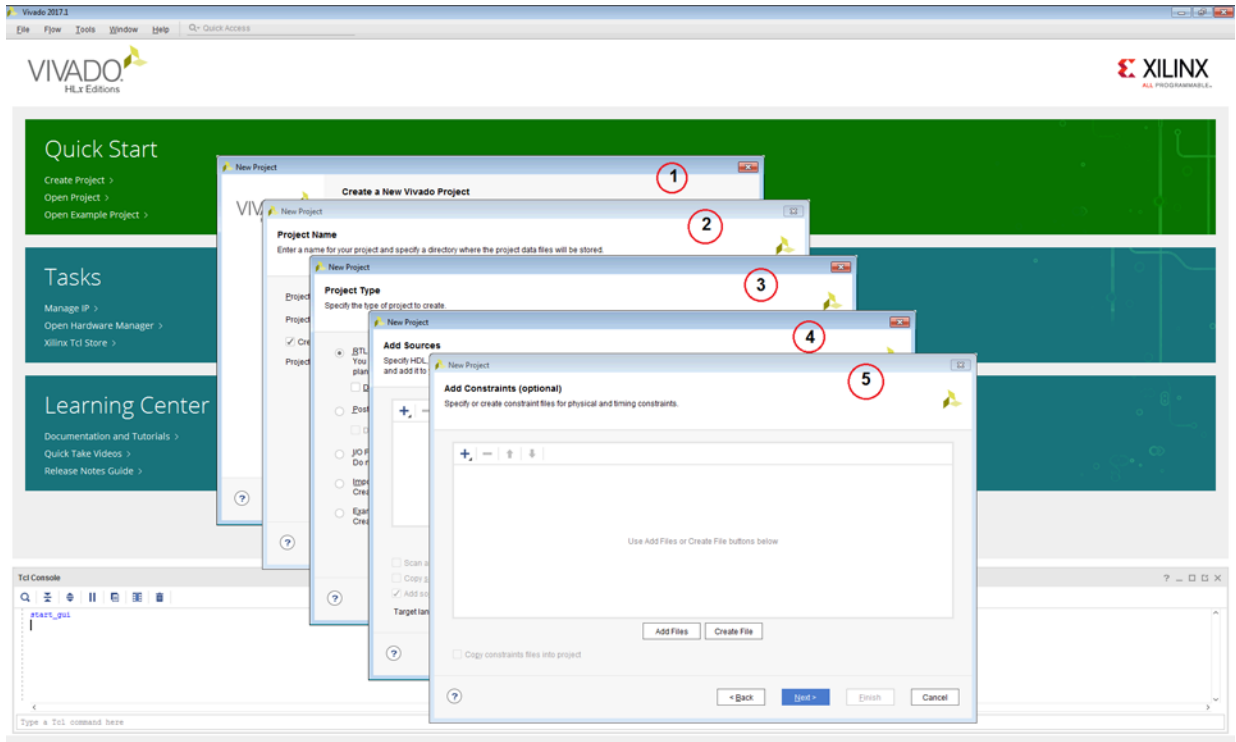
- **TX PLL Type:** Select the QPLL for TX UHD-SDI data path. Available options:
 - QPLL0
 - QPLL1
- **TX Ref Clock Selection:** Select the reference clock input source for TX transceiver QPLL. Available options:
 - GTREFCLK0
 - GTREFCLK1
 - GTNORTHREFCLK0
 - GTNORTHREFCLK1
 - GTSOUTHREFCLK0

- GTSOUTHREFCLK1
- **RX PLL Type:** Select the QPLL for RX UHD-SDI data path. Available options:
 - QPLL0
 - QPLL1
- **RX Ref Clock Selection:** Select the reference clock input source for RX transceiver QPLL. Available options:
 - GTREFCLK0
 - GTREFCLK1
 - GTNORTHREFCLK0
 - GTNORTHREFCLK1
 - GTSOUTHREFCLK0
 - GTSOUTHREFCLK1
- **DRP Clock Freq:** Select the DRP clock frequency in MHz.

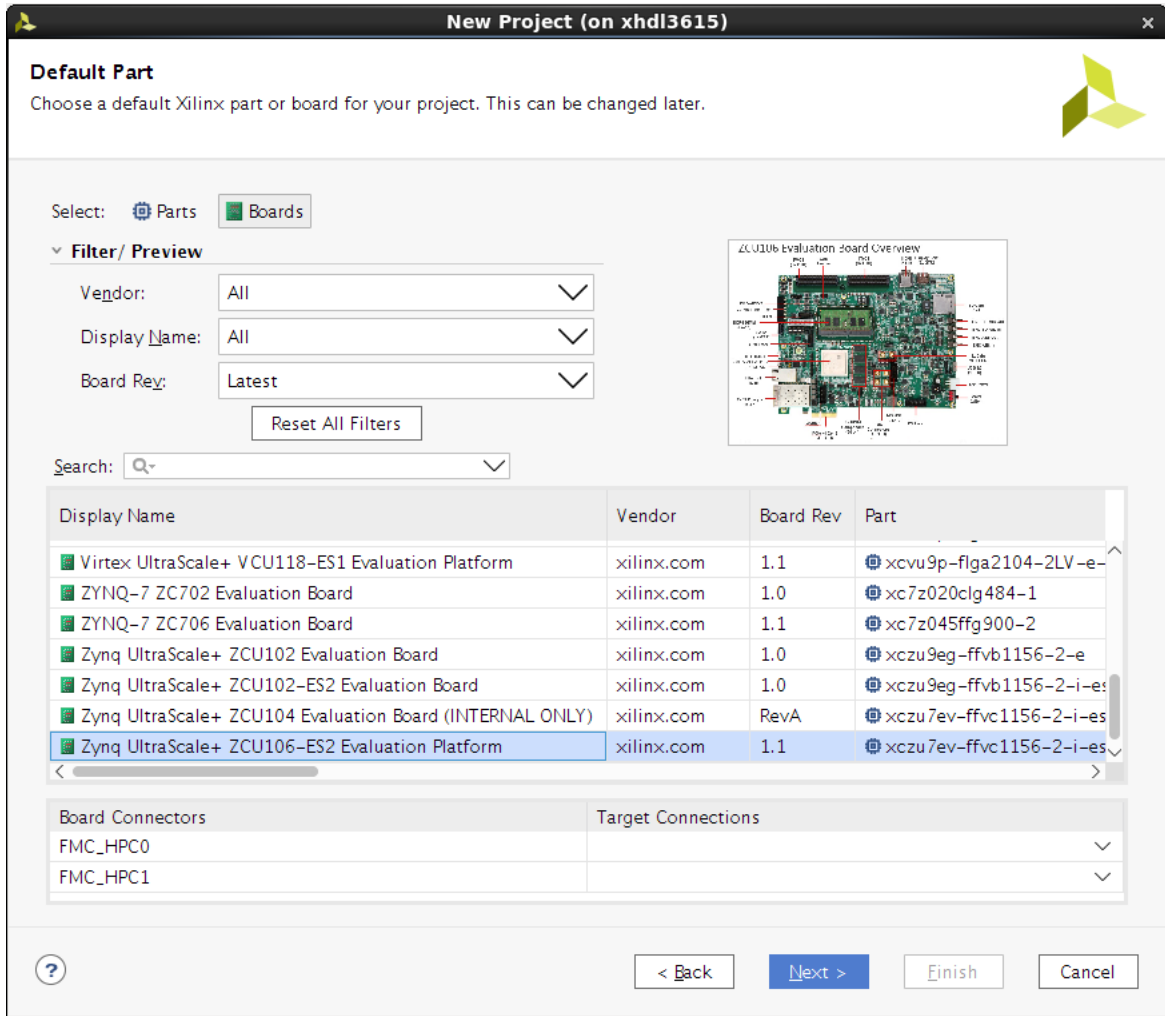
You can select the reference clock based on your board requirements and customize the `uhdsdi_gt` core. The `uhdsdi_gt` core generates sources files in Verilog which you can view and modify based on your use case.

Running the ZCU106 Example Design

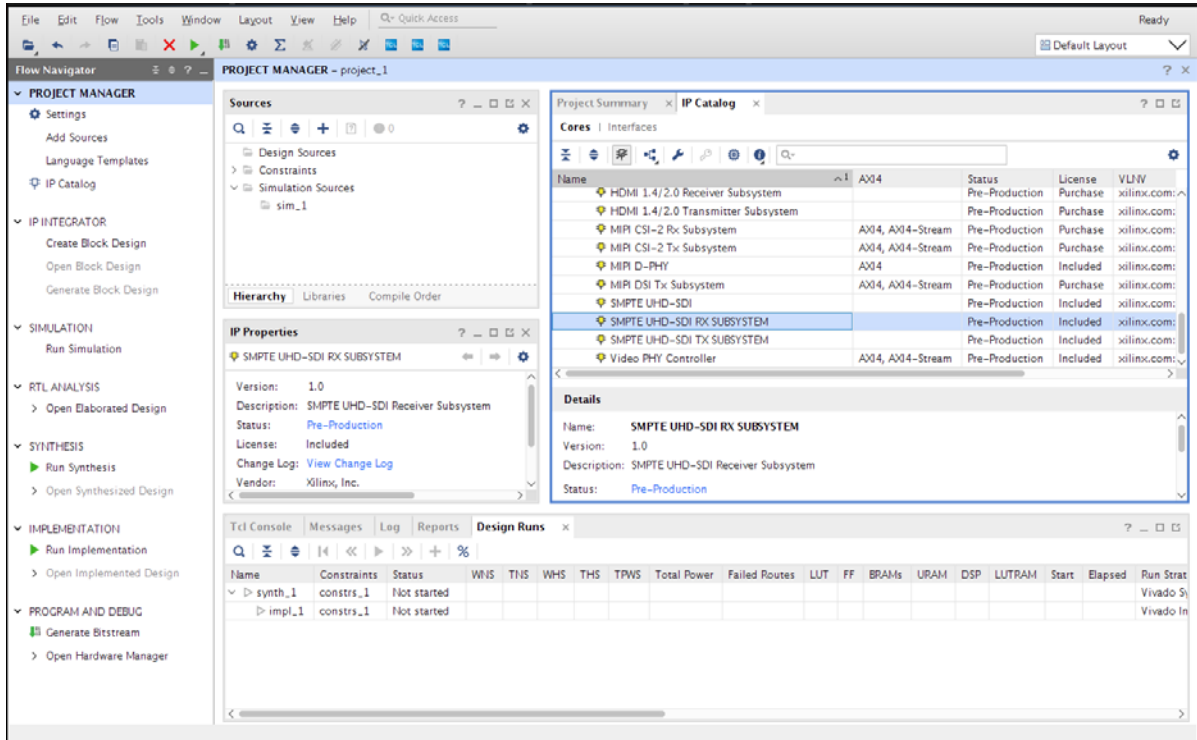
1. Open the Vivado Design Suite and create a new project.
2. In the pop-up window, press **Next** 5 times.



3. Select the Board. (ZCU106 supported.)



4. Click **Finish**.
5. Click **IP Catalog** and double-click **SMPT E UHD-SDI RX Subsystem** under Video Connectivity.



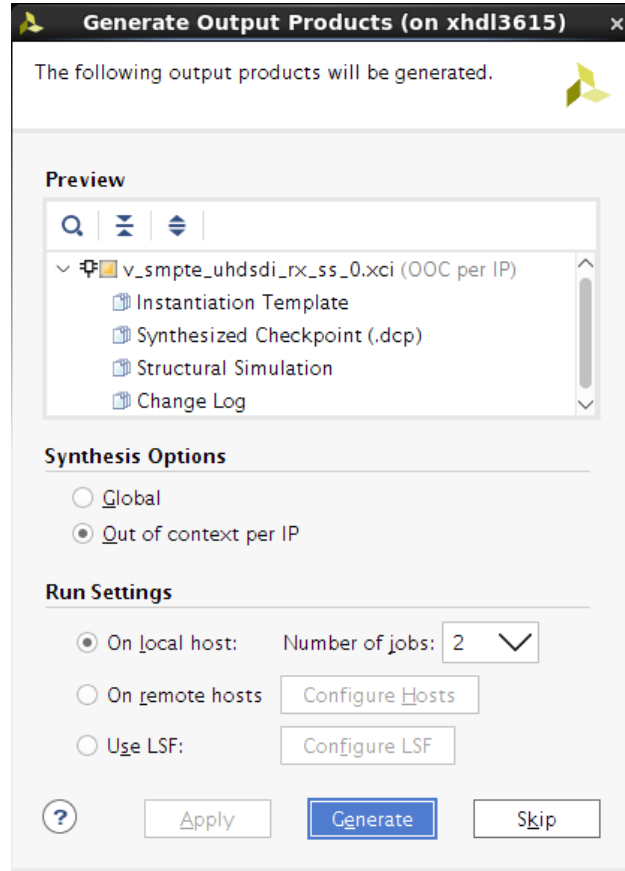
For the Application Example Design flow, IP configuration is based on options selected in **Application Example Design** tab.

You can rename the IP component name, which is used as Application Example Design project name.

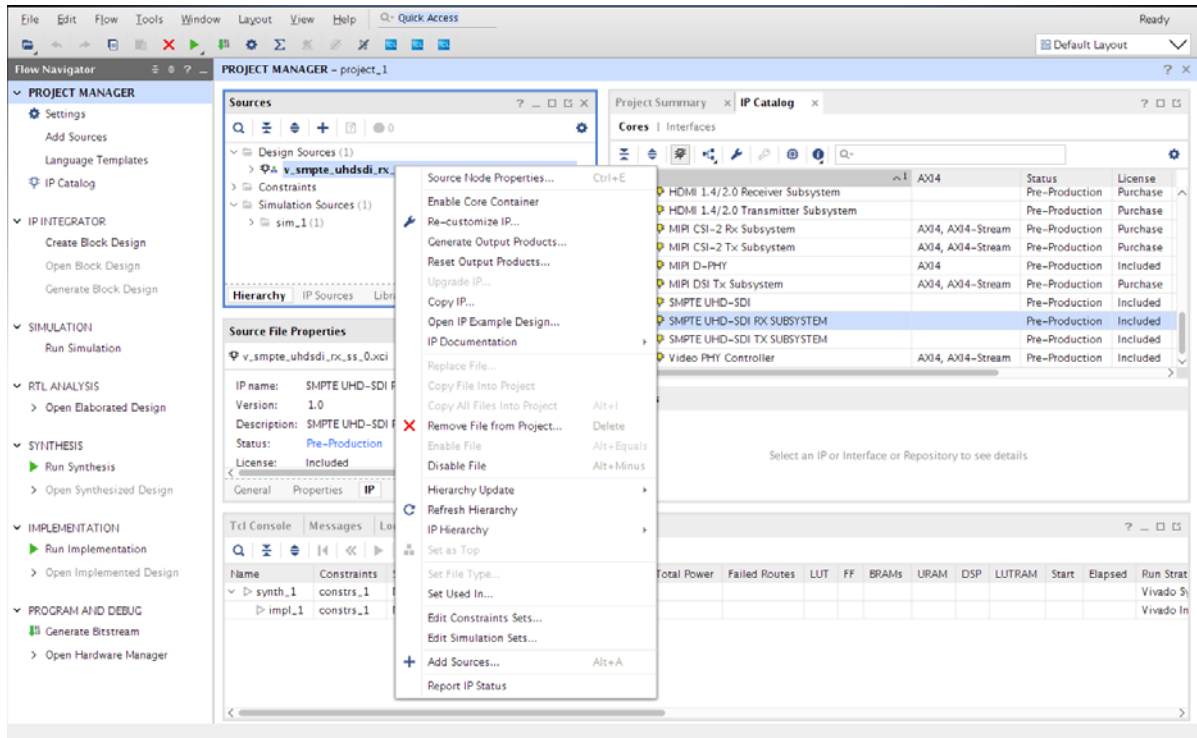
Configure SMPTE UHD-SDI RX Subsystem 'Application Example Design' tab, then click **OK**.

The Generate Output Products dialog box appears

6. Click **Generate**.
 - a. You may optionally click **Skip** if you only want to generate the Application Example Design.

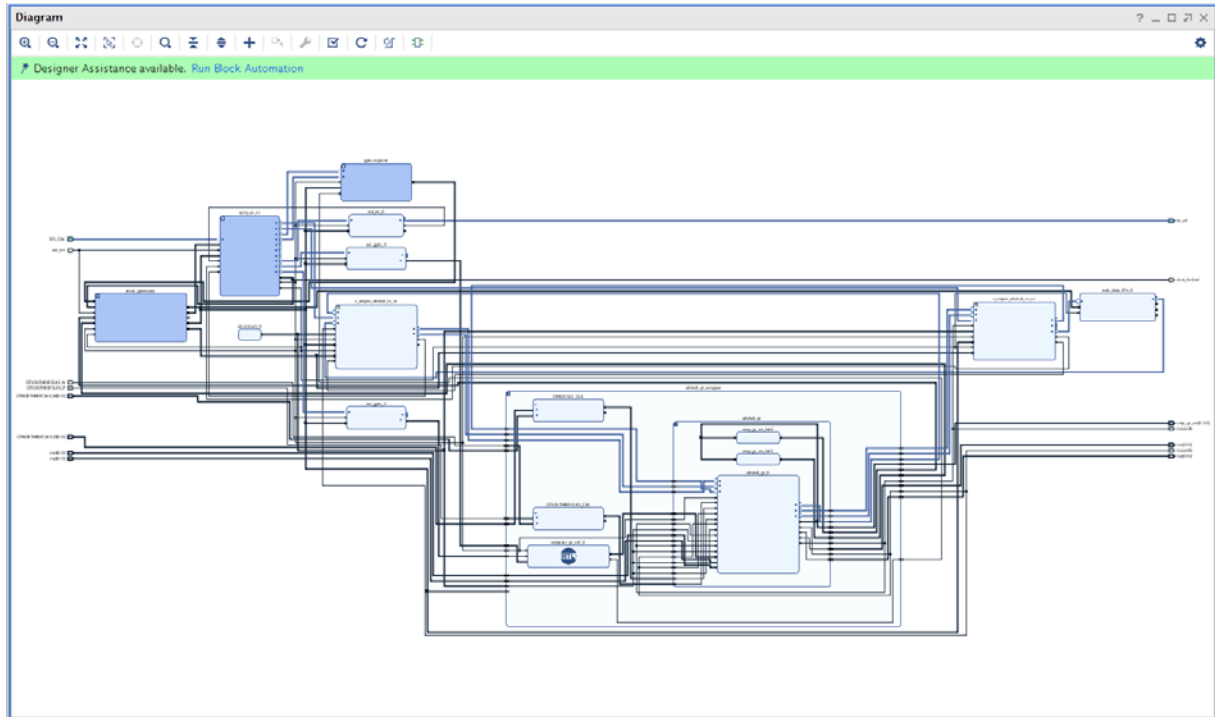


7. Right-click the SMPTE UHD-SDI RX Subsystem component under Design source, and click **Open IP Example Design**.



8. Choose the target project location, then click **OK**.

The IPI Design is then generated and creates the SDK application and generates an **.elf** file. You may choose to Run Synthesis, Implementation, or Generate Bitstream. An overall system IPI block diagram of the ZCU106 based Application Example Design is shown.



Requirements

Hardware

The hardware requirements for this reference system are:

- Xilinx Zynq UltraScale+ MPSoC ZCU106 Evaluation Kit
- SDI source equipment
- SDI sink equipment

Software

This section includes any software requirements:

- Vivado Design Suite 2017.3 or later
- SDK 2017.3 or later
- Software terminals (for example, Tera Term, HyperTerminal or PuTTY)

Setup

The reference design runs on the Zynq UltraScale+ MPSoC board (ZCU106) using SDI connectors available on the board.

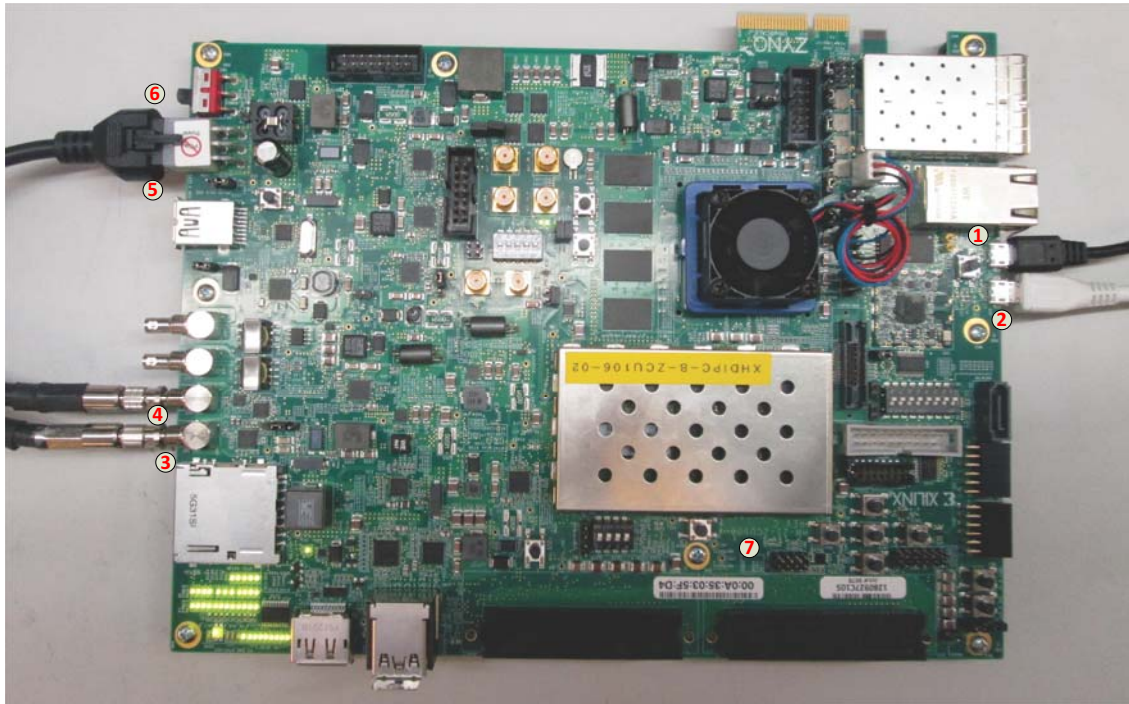


Figure 5-6: ZCU106 Board Setup

In the following procedure, the numbers in parentheses correspond to the callout numbers in [Figure 5-6](#).

1. Connect a USB cable from the host PC to the USB JTAG port (1). Ensure the appropriate device drivers are installed.
2. Connect a second USB cable from the host PC to the USB UART port (2). Ensure that the USB UART drivers described in [Hardware](#) have been installed.
3. Connect the SDI_INT link of ZCU106 (3) to the SDI source device.
4. Connect the SDI_OUT link of ZCU106 (4) to the SDI sink device.
5. Connect the ZCU106 board to a power supply slot (5).
6. Switch on the ZCU106 board (6).
7. Make sure that the HW-ZCU106 board revision (7) is Rev C.
8. Ensure that the SMPTE 352/Payload ID is enabled in the SDI Stream connected to the SDI input link of ZCU106.
9. Start Tera Term or PuTTY to connect to the COM port interface 0 on the Host PC with 115200 bps, 8 bits, No parity, 1 stop bit, and no flow control as configuration.

Compiling Software in SDK

The UHD-SDI Application Example Design generates an .elf file automatically. Use the following procedure if you want to open the SDK project for UHD-SDI example design from Vivado:

1. In Vivado, click **File -> Launch SDK**.
2. Select **Exported location** and **workspace** as **<Proj Dir>/<Component Name>_ex/<Component Name>_ex.sdk** and click **OK** to launch and open SDK project.

SDK project opens.

Running the Design on the Hardware

The following steps are used to run the BIT and ELF files on the hardware setup:

1. Connect the JTAG cable and USB-UART cable to the board.
2. Navigate to **<Component Name>_ex/imports**.
3. Start the Xilinx Software Debugger (XSDB) by sourcing xsdb from the build area from the command prompt.
4. Run the following command to program FPGA and to execute the application.

```
source xsdb.tcl
```

5. To observe the results, open Tera Term or PUTTY and configure its serial port (Interface 0) to 115200 baud with the default configuration. Make sure that the UART cable is connected to the board and the PC.

The UART console displays SDI stream details on console.

UART Console Screens

The following figure shows the initial UART console output along with menu options.

```
-----
SDI Pass Through Example   ---
(c) 2017 by Xilinx, Inc.   ---
-----
Build Sep 18 2017 - 01:48:20
-----

-----
MAIN MENU   ---
-----
i - Info
  => Shows information about the SDI RX stream, SDI TX stream.
z - SDI TX & RX log
  => Shows log information for SDI TX & RX.
d - Debug Info
  => Registers Dump.

INFO>> SDI Rx: Input Locked
-----
SDI TX SubSystem
-----
SDI stream info
-----
Color Format:      YUV_422
Color Depth:      10
Pixels Per Clock: 2
Mode:             Progressive
Frame Rate:       60Hz
Resolution:       4096x2160@60Hz
Pixel Clock:      594000000
SDI Mode:         12G
Bit Rate:         Integer
ST352 Payload:    0x140CBCE
```


The following figure shows the UART console output when **i** is pressed.

```

-----
Info
-----

-----
SDI TX SubSystem
-----

->SDI TX Subsystem Cores
 : SDI TX
 : VTC Core

SDI stream info
-----
Color Format:      YUV_422
Color Depth:      10
Pixels Per Clock: 2
Mode:             Progressive
Frame Rate:       60Hz
Resolution:       4096x2160@60Hz
Pixel Clock:      594000000
SDI Mode:         12G
Bit Rate:         Integer
ST352 Payload:   0x140CBCE
No Error Detected

SDI TX timing
-----

HSYNC Timing: hav=4096, hfp=88, hsw=88(hsp=1), hbp=128, htot=4400
VSYNC Timing: vav=2160, vfp=08, vsw=10(vsp=1), vbp=072, vtot=2250

-----

SDI Rx SubSystem
-----

->SDI RX Subsystem Cores
 : SDI RX

SDI stream info
-----
Color Format:      YUV_422
Color Depth:      10
Pixels Per Clock: 2
Mode:             Progressive
Frame Rate:       60Hz
Resolution:       4096x2160@60Hz
Pixel Clock:      594000000
SDI Mode:         12G

```

The following screen appears in the console output when **z** is pressed.

```
SDI TX log
-----
Initializing SDI TX core....
Initializing VTC core....
Configure SDI TX Core....
TX Stream is Up
TX Stream Start

SDI RX log
-----
Initializing SDI RX core....
RX Stream is Up
```

The following screen appears in the console output when **d** is pressed:

```

-----
Info
-----

-----
SDI TX SubSystem
-----

Debug info
-----
TX Video Bridge:
  Bridge Select: 12G SDI Bridge
  3G Bridge SDI Mode: HD

TX AXIS Bridge:
  Locked: 1
  Overflow: 0
  Underflow: 1

SDI Registers Dump
-----
Address: 0x80020000 Data: 0x301
Address: 0x80020004 Data: 0x117250
Address: 0x80020008 Data: 0x0
Address: 0x8002000C Data: 0x1
Address: 0x80020010 Data: 0x500
Address: 0x80020014 Data: 0x1
Address: 0x80020018 Data: 0x23C000A
Address: 0x8002001C Data: 0x140CBCE
Address: 0x80020020 Data: 0x140CBCE
Address: 0x80020024 Data: 0x140CBCE
Address: 0x80020028 Data: 0x140CBCE
Address: 0x8002002C Data: 0x0
Address: 0x80020030 Data: 0x0
Address: 0x80020034 Data: 0x0
Address: 0x80020038 Data: 0x0
Address: 0x8002003C Data: 0x1000000
Address: 0x80020040 Data: 0x2
Address: 0x80020044 Data: 0x0
Address: 0x80020048 Data: 0x0
Address: 0x8002004C Data: 0x0
Address: 0x80020050 Data: 0x0
Address: 0x80020054 Data: 0x0
Address: 0x80020058 Data: 0x0
Address: 0x8002005C Data: 0x0
Address: 0x80020060 Data: 0x30000E5
Address: 0x80020064 Data: 0x0
Address: 0x80020068 Data: 0x1
Address: 0x8002006C Data: 0xC0186

```

Verification, Compliance, and Interoperability

The SMPTE UHD-SDI RX Subsystem has been verified using both simulation and hardware testing.

A highly parameterizable transaction-based simulation test suite has been used to verify the subsystem. The tests include:

- Different SDI standard.
- Different resolutions with different video timing parameters.
- Recovery from error conditions.
- Register read and write access.

Interoperability

The SMPTE UHD-SDI Receiver Subsystem has been tested using standard off-the-shelf SDI test equipment with a variety of UHD-SDI devices. It is compliant with the SMPTE SDI standards.

Hardware Validation

The SMPTE UHD-SDI Receiver Subsystem is tested in hardware for functionality, performance, and reliability using Xilinx® evaluation platforms. The SMPTE UHD-SDI Receiver Subsystem verification test suites for all possible modules are continuously being updated to increase test coverage across the range of possible parameters for each individual module.

The SMPTE UHD-SDI Receiver Subsystem has been validated using

- Zynq® UltraScale+™ MPSoC ZCU106 Evaluation Kit
- Zynq® UltraScale+™ MPSoC ZCU102 Evaluation Kit

The SMPTE UHD-SDI Receiver Subsystem is tested with following devices:

- Phabrix QX 12G as Source and Sync device
- Phabrix R1000 as Source and Sync device
- Phabrix SX as Source and Sync device
- Omnitek Ultra 4K Tool box as Source and Sync device

Video Resolutions

A series of interoperability test scenarios, listed in [Table A-1](#), are validated for different resolutions. Test equipment is used to drive the SDI mode specific traffic and display the received SDI data. [Figure A-1](#) shows the UHD-SDI validation setup using SDI Test equipment.

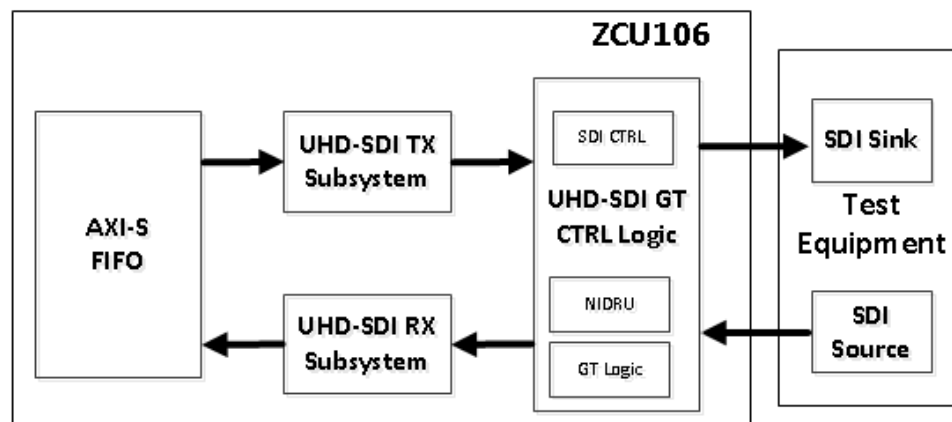


Figure A-1: UHD-SDI Test Equipment Setup

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
HD-SDI	1920x1080i50	Phabrix QX 12G	Phabrix QX 12G	2640	1920	1125	1080	50
HD-SDI	1920x1080i59.94	Phabrix QX 12G	Phabrix QX 12G	2200	1920	1125	1080	59.94
HD-SDI	1920x1080i60	Phabrix QX 12G	Phabrix QX 12G	2200	1920	1125	1080	60
HD-SDI	1280x720p25	Phabrix QX 12G	Phabrix QX 12G	3960	1280	750	720	25
HD-SDI	1280x720p29.97	Phabrix QX 12G	Phabrix QX 12G	3300	1280	750	720	29.97
HD-SDI	1280x720p30	Phabrix QX 12G	Phabrix QX 12G	3300	1280	750	720	30
HD-SDI	1280x720p50	Phabrix QX 12G	Phabrix QX 12G	1980	1280	750	720	50
HD-SDI	1280x720p59.94	Phabrix QX 12G	Phabrix QX 12G	1650	1280	750	720	59.94
HD-SDI	1280x720p60	Phabrix QX 12G	Phabrix QX 12G	1650	1280	750	720	60
HD-SDI	1920x1080p23.98	Phabrix QX 12G	Phabrix QX 12G	2750	1920	1125	1080	23.98
HD-SDI	1920x1080p24	Phabrix QX 12G	Phabrix QX 12G	2750	1920	1125	1080	24
HD-SDI	1920x1080p25	Phabrix QX 12G	Phabrix QX 12G	2640	1920	1125	1080	25
HD-SDI	1920x1080p29.97	Phabrix QX 12G	Phabrix QX 12G	2200	1920	1125	1080	29.97
HD-SDI	1920x1080p30	Phabrix QX 12G	Phabrix QX 12G	2200	1920	1125	1080	30
3G-SDI Level A	1920x1080p50	Phabrix QX 12G	Phabrix QX 12G	2640	1920	1125	1080	50
3G-SDI Level A	1920x1080p59.94	Phabrix QX 12G	Phabrix QX 12G	2200	1920	1125	1080	59.94
3G-SDI Level A	1920x1080p60	Phabrix QX 12G	Phabrix QX 12G	2200	1920	1125	1080	60
HD-SDI	2048x1080p23.98	Phabrix QX 12G	Phabrix QX 12G	2750	2048	1125	1080	23.98
HD-SDI	2048x1080p24	Phabrix QX 12G	Phabrix QX 12G	2750	2048	1125	1080	24
HD-SDI	2048x1080p25	Phabrix QX 12G	Phabrix QX 12G	2640	2048	1125	1080	25
HD-SDI	2048x1080p29.97	Phabrix QX 12G	Phabrix QX 12G	2200	2048	1125	1080	29.97
HD-SDI	2048x1080p30	Phabrix QX 12G	Phabrix QX 12G	2200	2048	1125	1080	30
3G-SDI Level A, 3G-SDI Level B	2048x1080p47.95	Phabrix QX 12G	Phabrix QX 12G	2750	2048	1125	1080	47.95
3G-SDI Level A, 3G-SDI Level B	2048x1080p48	Phabrix QX 12G	Phabrix QX 12G	2750	2048	1125	1080	48

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
3G-SDI LevelA, 3G-SDI Level B	2048x1080p50	Phabrix QX 12G	Phabrix QX 12G	2640	2048	1125	1080	50
3G-SDI LevelA, 3G-SDI Level B	2048x1080p59.94	Phabrix QX 12G	Phabrix QX 12G	2200	2048	1125	1080	59.94
3G-SDI LevelA, 3G-SDI Level B	2048x1080p60	Phabrix QX 12G	Phabrix QX 12G	2200	2048	1125	1080	60
6G-SDI	3840x2160p23.98	Phabrix QX 12G	Phabrix QX 12G	5500	3840	2250	2160	23.98
6G-SDI	3840x2160p24	Phabrix QX 12G	Phabrix QX 12G	5500	3840	2250	2160	24
6G-SDI	3840x2160p25	Phabrix QX 12G	Phabrix QX 12G	5280	3840	2250	2160	25
6G-SDI	3840x2160p29.97	Phabrix QX 12G	Phabrix QX 12G	4400	3840	2250	2160	29.97
6G-SDI	3840x2160p30	Phabrix QX 12G	Phabrix QX 12G	4400	3840	2250	2160	30
12G-SDI	3840x2160p50	Phabrix QX 12G	Phabrix QX 12G	5280	3840	2250	2160	50
12G-SDI	3840x2160p59.94	Phabrix QX 12G	Phabrix QX 12G	4400	3840	2250	2160	59.94
12G-SDI	3840x2160p60	Phabrix QX 12G	Phabrix QX 12G	4400	3840	2250	2160	60
6G-SDI	4096x2160p23.98	Phabrix QX 12G	Phabrix QX 12G	5500	4096	2200	2160	23.98
6G-SDI	4096x2160p24	Phabrix QX 12G	Phabrix QX 12G	5500	4096	2200	2160	24
6G-SDI	4096x2160p25	Phabrix QX 12G	Phabrix QX 12G	5280	4096	2250	2160	25
6G-SDI	4096x2160p29.97	Phabrix QX 12G	Phabrix QX 12G	4400	4096	2250	2160	29.97
6G-SDI	4096x2160p30	Phabrix QX 12G	Phabrix QX 12G	4400	4096	2250	2160	30
12G-SDI	4096x2160p47.95	Phabrix QX 12G	Phabrix QX 12G	5500	4096	2250	2160	47.95
12G-SDI	4096x2160p48	Phabrix QX 12G	Phabrix QX 12G	5500	4096	2250	2160	48
12G-SDI	4096x2160p50	Phabrix QX 12G	Phabrix QX 12G	5280	4096	2250	2160	50
12G-SDI	4096x2160p59.94	Phabrix QX 12G	Phabrix QX 12G	4400	4096	2250	2160	59.94
12G-SDI	4096x2160p60	Phabrix QX 12G	Phabrix QX 12G	4400	4096	2250	2160	60
SD-SDI	625i50	Phabrix R1000, Phabrix SX	Phabrix R1000, Phabrix SX	864	720	625	576	50
HD-SDI	1920x1080sF23.98	Phabrix R1000, Phabrix SX	Phabrix R1000, Phabrix SX	2750	1920	1125	1080	23.98
HD-SDI	1920x1080sF24	Phabrix R1000, Phabrix SX	Phabrix R1000, Phabrix SX	2750	1920	1125	1080	23.98

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
HD-SDI	1920x1080sF25	Phabrix R1000, Phabrix SX	Phabrix R1000, Phabrix SX	2640	1920	1125	1080	23.98
HD-SDI	1920x1080sF29.97	Phabrix R1000, Phabrix SX	Phabrix R1000, Phabrix SX	2200	1920	1125	1080	29.97
HD-SDI	1920x1080sF30	Phabrix R1000, Phabrix SX	Phabrix R1000, Phabrix SX	2200	1920	1125	1080	30
3G-SDI Level A, 3G-SDI Level B	1920x1080p47.95	Phabrix R1000, Phabrix SX	Phabrix R1000, Phabrix SX	2750	1920	1125	1080	47.95
3G-SDI Level A, 3G-SDI Level B	1920x1080p48	Phabrix R1000, Phabrix SX	Phabrix R1000, Phabrix SX	2750	1920	1125	1080	48
SD-HDI	720x576i50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	720	-	576	50
HD-SDI	1920x1080i47.95	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	47.95
HD-SDI	1920x1080i48	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	48
HD-SDI	1920x1080i50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	50
HD-SDI	1920x1080i59.94	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	59.94
HD-SDI	1920x1080i60	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	60

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
HD-SDI	2048x1080i47.95	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	47.95
HD-SDI	2048x1080i48	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	48
HD-SDI	2048x1080i50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	50
HD-SDI	2048x1080i59.94	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	59.94
HD-SDI	2048x1080i60	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	60
HD-SDI	1280x720p25	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1280	-	720	25
HD-SDI	1280x720p29.97	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1280	-	720	29.97
HD-SDI	1280x720p30	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1280	-	720	30
HD-SDI	1280x720p50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1280	-	720	50
HD-SDI	1280x720p59.94	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1280	-	720	59.94

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
HD-SDI	1280x720p60	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1280	-	720	60
HD-SDI	1920x1080p23.98	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	23.98
HD-SDI	1920x1080p24	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	24
HD-SDI	1920x1080p25	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	25
HD-SDI	1920x1080p29.97	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	29.97
HD-SDI	1920x1080p30	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	30
HD-SDI	1920x1080sF23.98	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	23.98
HD-SDI	1920x1080sF24	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	24
HD-SDI	1920x1080sF25	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	25
HD-SDI	1920x1080sF29.97	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	29.97

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
HD-SDI	1920x1080sF30	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	30
HD-SDI	2048x1080p23.98	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	23.98
HD-SDI	2048x1080p24	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	24
HD-SDI	2048x1080p25	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	25
HD-SDI	2048x1080p29.97	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	29.97
HD-SDI	2048x1080p30	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	30
HD-SDI	2048x1080sF23.98	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	23.98
HD-SDI	2048x1080sF24	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	24
HD-SDI	2048x1080sF25	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	25
HD-SDI	2048x1080sF29.97	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	29.97

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
HD-SDI	2048x1080sF30	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	30
3G-A	1920x1080p47.95	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	47.95
3G-A	1920x1080p48	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	48
3G-A	1920x1080p50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	50
3G-A	1920x1080p59.94	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	59.94
3G-A	1920x1080p60	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	60
3G-A	2048x1080p47.95	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	47.95
3G-A	2048x1080p48	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	48
3G-A	2048x1080p50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	50
3G-A	2048x1080p59.94	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	59.94

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
3G-A	2048x1080p60	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	60
3G-B DL	1920x1080p47.95	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	47.95
3G-B DL	1920x1080p48	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	48
3G-B DL	1920x1080p50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	50
3G-B DL	1920x1080p59.94	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	59.94
3G-B DL	1920x1080p60	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	60
3G-B DL	2048x1080p47.95	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	47.95
3G-B DL	2048x1080p48	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	48
3G-B DL	2048x1080p50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	50
3G-B DL	2048x1080p59.94	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	59.94

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
3G-B DL	2048x1080p60	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	60
3G-B DS	1920x1080i47.95	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	47.95
3G-B DS	1920x1080i48	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	48
3G-B DS	1920x1080i50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	50
3G-B DS	1920x1080i59.94	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	59.94
3G-B DS	1920x1080i60	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	60
3G-B DS	2048x1080i47.95	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	47.95
3G-B DS	2048x1080i48	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	48
3G-B DS	2048x1080i50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	50
3G-B DS	2048x1080i59.94	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	59.94

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
3G-B DS	2048x1080i60	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	60
3G-B DS	1920x1080sF23.98	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	23.98
3G-B DS	1920x1080sF24	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	24
3G-B DS	1920x1080sF25	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	25
3G-B DS	1920x1080sF29.97	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	29.97
3G-B DS	1920x1080sF30	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	1920	-	1080	30
3G-B DS	2048x1080sF23.98	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	23.98
3G-B DS	2048x1080sF24	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	24
3G-B DS	2048x1080sF25	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	25
3G-B DS	2048x1080sF29.97	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	29.97

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
3G-B DS	2048x1080sF30	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	2048	-	1080	30
6G-SDI	3840x2160p23.98	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	3840	-	2160	23.98
6G-SDI	3840x2160p24	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	3840	-	2160	24
6G-SDI	3840x2160p25	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	3840	-	2160	25
6G-SDI	3840x2160p29.97	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	3840	-	2160	29.97
6G-SDI	3840x2160p30	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	3840	-	2160	30
6G-SDI	4096x2160p23.98	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	4096	-	2160	23.98
6G-SDI	4096x2160p24	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	4096	-	2160	24
6G-SDI	4096x2160p25	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	4096	-	2160	25
6G-SDI	4096x2160p29.97	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	4096	-	2160	29.97

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
6G-SDI	4096x2160p30	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	4096	-	2160	30
12G-SDI	3840x2160p47.95	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	3840	-	2160	47.95
12G-SDI	3840x2160p48	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	3840	-	2160	48
12G-SDI	3840x2160p50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	3840	-	2160	50
12G-SDI	3840x2160p59.94	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	3840	-	2160	59.94
12G-SDI	3840x2160p60	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	3840	-	2160	60
12G-SDI	4096x2160p47.95	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	4096	-	2160	47.95
12G-SDI	4096x2160p48	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	4096	-	2160	48
12G-SDI	4096x2160p50	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	4096	-	2160	50

Table A-1: Tested Video Resolutions for YCbCr 4:2:2 at 10 Bits/Component (Cont'd)

SDI Mode	Resolution	SDI Source	SDI Sink	Horizontal		Vertical		Frame Rate (Hz)
				Total	Active	Total	Active	
12G-SDI	4096x2160p59.94	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	4096	-	2160	59.94
12G-SDI	4096x2160p60	Omnitek Ultra 4K Tool box	Omnitek Ultra 4K Tool box	-	4096	-	2160	60

Debugging

This appendix includes details about resources available on the Xilinx Support website and debugging tools.

Finding Help on Xilinx.com

To help in the design and debug process when using the SMPTE UHD-SDI, the [Xilinx Support web page](http://www.xilinx.com/support) (www.xilinx.com/support) contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support.

Documentation

This product guide is the main document associated with the SMPTE UHD-SDI. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page (www.xilinx.com/support) or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the Design Tools tab on the Downloads page (www.xilinx.com/download). For more information about this tool and the features available, open the online help after installation.

Solution Centers

See the [Xilinx Solution Centers](#) for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

The Solution Center specific to the SMPTE UHD-SDI core is listed below.

- [Xilinx Video Solution Center](#)

Answer Records

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product.

Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

Answer Records for this core can be located by using the Search Support box on the main [Xilinx support web page](#). To maximize your search results, use proper keywords such as

- Product name
- Tool message(s)
- Summary of the issue encountered

A filter search is available after results are returned to further target the results.

Master Answer Record for the SMPTE UHD-SDI

AR: [68766](#)

Technical Support

Xilinx provides technical support at the [Xilinx Support web page](#) for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support if you do any of the following:

- Implement the solution in devices that are not defined in the documentation.
- Customize the solution beyond that allowed in the product documentation.
- Change any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support, navigate to the [Xilinx Support web page](#).

Debug Tools

There are many tools available to address SMPTE UHD-SDI design issues. It is important to know which tools are useful for debugging various situations.

Vivado Lab Edition

Vivado® Lab Edition inserts logic analyzer and virtual I/O cores directly into your design. Vivado Lab Edition also allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx.

The Vivado logic analyzer is used with the logic debug IP cores, including:

- ILA 2.0 (and later versions)
- VIO 2.0 (and later versions)

See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 6].

Hardware Debug

Hardware issues can range from link bring-up to problems seen after hours of testing. This section provides debug steps for common issues. The Vivado debug feature is a valuable resource to use in hardware debug. The signal names mentioned in the following individual sections can be probed using the debug feature for debugging the specific problems.

General Checks

Ensure that all the timing constraints for the core were properly incorporated from the example design and that all constraints were met during implementation.

- Check that MMCM lock and PLL lock signal(s) are asserted.
- Verify the IO pin planning and XDC constraints.
- Follow the recommended reset sequence.
- Verify all clocks are connected and are with expected frequencies.
- Enable AXI-4 Lite based register interface to get core status and control.
- Make sure serial line trace lengths are equal
- Verify the FMC_VADJ voltage to 1.8v in case of FMC card usage.

Figure B-1 shows the steps to perform a hardware debug.

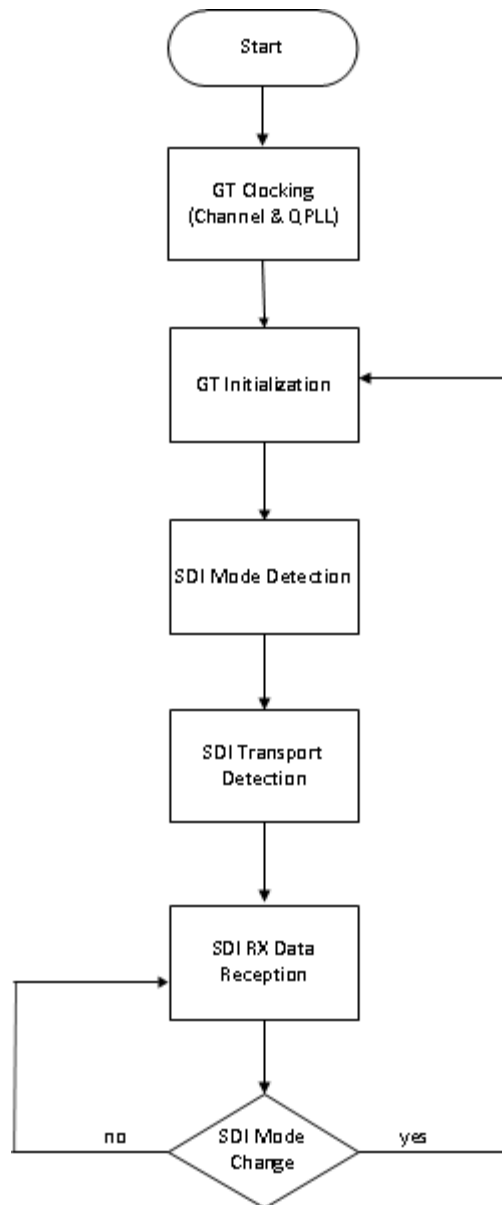


Figure B-1: Hardware Debugging

GT Clocking

- Make sure QPLL is getting reset before starting the IP.
- Monitor the QPLL LOCK signal.
- Verify that QPLL input clock frequency is of expected value.
- It is mandatory to reset the QPLL if clock input to QPLL is stopped or unstable.
- Refer the AR#[57738](#) for debugging GT reference clock issues.

- Make sure to use QPLL default settings from latest GT Wizard IP core based on target device.
- Check the voltage rails on the transceivers. Refer AR#[57737](#) for more information.
- Measure RXOUTCLK is of expected frequency.
- Make sure RXOUTCLK of the transceiver is the clock driving rx_usrclk, RXUSRCLK, and RXUSRCLK2.
- Monitor RXBUFFSTATUS[2:0] for overflow and underflow errors.

GT Initialization

- GTRXRESETDONE is asserted High after GT completes initialization.
- Make sure GT is not reset during normal operation.
- Refer AR#[59435](#) for more information on debugging GT reset problems.
- Follow recommended GT reset sequence.

SDI Mode Detection

- UHD-SDI RX core hunts for TRS symbols and asserts rx_mode_locked if UHD-SDI core detects error free TRS symbols for continuous three video lines.
- Look for pulses on the rx_eav, rx_sav signals at the correct places on each line along with rx_trs signal.
- Make sure no CRC errors are reported on rx_crc_err_ds1 to ds16.
- Make sure supported modes are enabled in rx_mode_enable[5:0] signal.
- Verify that rx_mode[2:0] reported by SDI core is matching with incoming SDI mode.
- SDI Mode detection will search the mode in the following fashion if all of the modes are set in rx_mode_enable : HD -> 3G -> 6G -> 12GA -> 12GB -> SD -> HD.
- Make sure that rx_frame_en signal is asserted high during core operation.

SDI Transport Detection

- UHD-SDI core reports the video format by asserting rx_t_locked signal.
- This functionality takes two video frames to report the transport format
- Check rx_t_family[3:0], rx_t_rate[3:0], rx_t_scan signals along with rx_t_locked signal.
- Make sure incoming video format is supported by UHD-SDI core.
- Check rx_bit_rate is connected to the core.

UHD-SDI RX Data Reception

- UHD-SDI core receives the data in this stage until it receives erroneous video lines or change in incoming video source.

UHD-SDI Mode Change

- UHD-SDI core deasserts `rx_mode_locked` signal if it receives 15 or more erroneous video lines continuously.
- After `rx_mode_locked` is deasserted, it starts the SDI mode scan.

SDI RX to Video Bridge Debug:

- Make sure correct SDI mode (`rx_mode` [2:0] input) is passed on to SDI RX to Video Bridge subcore.
- Verify first pixel data is data on input native SDI data stream is appearing on native Video interface by monitoring `vid_data[59:0]` data bus along with other control signals such as `vid_active_video`, `vid_hblank` and `vid_vblank`.

Video In to AXI4-Stream Debug:

- Verify first pixel data is data on input native video interface is appearing on AXI-4 Video interface by monitoring `m_axis_video_tdata[63:0]` data bus along with control signals such as `m_axis_video_tvalid` and `m_axis_video_tready`.
- Make sure that underflow and overflow outputs are not asserted. If asserted, verify the clock connections and along with expected frequencies.

Interface Debug

AXI4-Lite Interfaces

Read from a register that does not have all 0s as a default to verify that the interface is functional. See [Figure B-2](#) and [Figure B-3](#). Output `s_axi_arready` asserts when the read address is valid, and output `s_axi_rvalid` asserts when the read data/response is valid. If the interface is unresponsive, ensure that the following conditions are met:

- The `s_axi_aclk` and `aclk` inputs are connected and toggling.
- The interface is not being held in reset, and `s_axi_areset` is an active-Low reset.
- The interface is enabled, and `s_axi_aclken` is active-High (if used).
- The main core clocks are toggling and that the enables are also asserted.

- If the simulation has been run, verify in simulation and/or a debug feature capture that the waveform is correct for accessing the AXI4-Lite interface.

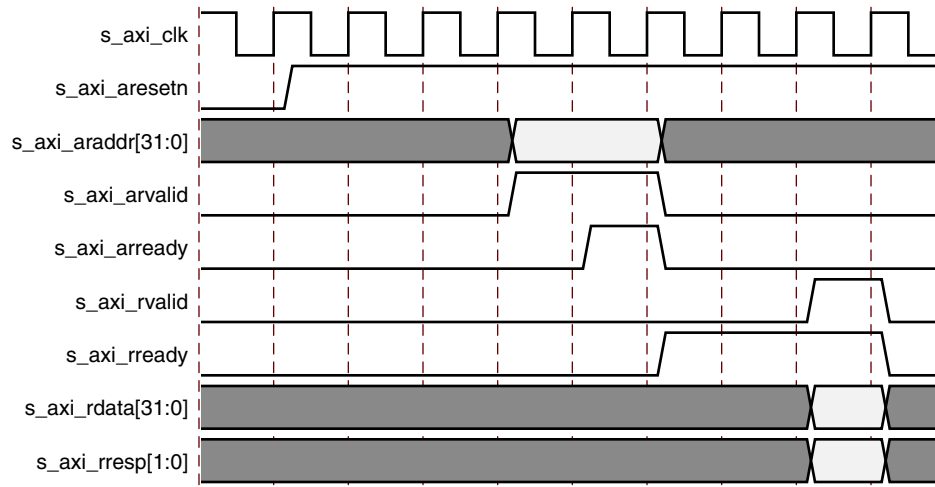


Figure B-2: Read

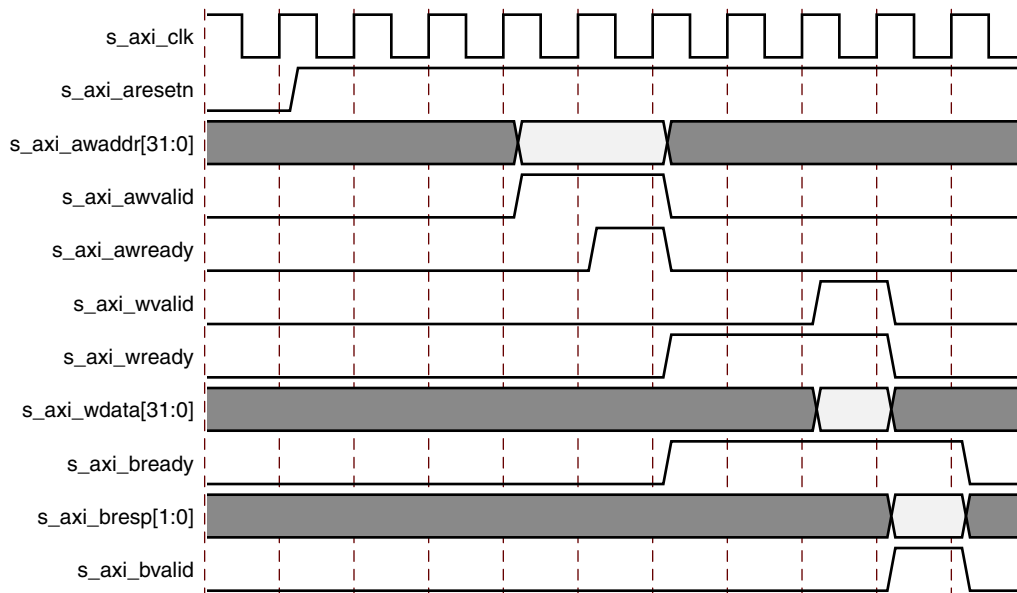


Figure B-3: Write

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see [Xilinx Support](#).

Documentation Navigator and Design Hubs

Xilinx Documentation Navigator provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open the Xilinx Documentation Navigator (DocNav):

- From the Vivado IDE, select **Help > Documentation and Tutorials**.
- On Windows, select **Start > All Programs > Xilinx Design Tools > DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the [Design Hubs](#) page.

Note: For more information on Documentation Navigator, see the [Documentation Navigator](#) page on the Xilinx website.

References

These documents provide supplemental material useful with this product guide:

1. *Vivado® Design Suite User Guide: Designing IP Subsystems using IP Integrator* ([UG994](#))
2. *Vivado Design Suite User Guide: Designing with IP* ([UG896](#))
3. *Vivado Design Suite User Guide: Getting Started* ([UG910](#))
4. *Vivado Design Suite User Guide: Logic Simulation* ([UG900](#))
5. *ISE® to Vivado Design Suite Migration Guide* ([UG911](#))
6. *Vivado Design Suite User Guide: Programming and Debugging* ([UG908](#))
7. *Vivado Design Suite User Guide - Implementation* ([UG904](#))
8. *SMPTE UHD-SDI Product Guide* ([PG0205](#))
9. *Video In to AXI-4 Stream LogiCORE IP Product Guide* ([PG043](#))
10. *AXI4-Stream Video IP and System Design Guide* ([UG934](#))
11. *UltraScale Architecture GTH Transceivers User Guide* ([UG576](#))
12. *UltraScale FPGAs Transceivers Wizard* ([PG182](#))
13. *Kintex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics* ([DS922](#))
14. *SMPTE UHD-SDI TX Subsystem Product Guide* ([PG289](#))
15. *Vivado Design Suite: AXI Reference Guide* ([UG1037](#))

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/04/2017	1.0	Initial Xilinx release.

Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special,

incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

© Copyright 2017 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.