SMPTE UHD-SDI Transmitter Subsystem v2.0

Product Guide

Vivado Design Suite

PG289 October 19, 2022

Xilinx is creating an environment where employees, customers, and partners feel welcome and included. To that end, we're removing non-inclusive language from our products and related collateral. We've launched an internal initiative to remove language that could exclude people or reinforce historical biases, including terms embedded in our software and IPs. You may still find examples of non-inclusive language in our older products as we work to make these changes and align with evolving industry standards. Follow this link for more information.

AMD
XILINX

AMD XILINX

Table of Contents

Introduction

The Society of Motion Picture and Television Engineers (SMPTE) UHD-SDI transmitter subsystem implements a serial digital interface (SDI) transmit interface in accordance with the SDI family of standards. The subsystem accepts video from an AXI4-Stream video interface and outputs a native video stream. It allows fast selection of 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

- Supports AXI4-Stream, native video and native SDI user interfaces
- Support for 2 pixels per sample
- 10-bit and 12-bit per color component
- Supports YCbCr 4:4:4, YCbCr 4:2:2, and YCbCr 4:2:0 color space
- Provision to insert ancillary data
- Supports HLG HDR video
- SMPTE 2081-10 HFR support in native SDI mode.
- SMPTE 2081-10 HFR support in AXI configurations for SDI TX subsystem: supports 6G and 12G 10-bit SDI mode.
- Supports block automation for Versal® ACAP device family
- AXI4-Lite interface for register access to configure different subsystem options
- SMPTE ST 352: Insertion of payload packets into Y Stream and C Stream are supported.
- Standards compliance:
  - SMPTE ST 259: SD-SDI at 270 Mb/s
  - 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 transmitter subsystems.

LogiCORE IP Facts Table	
Subsystem Specifics	
Supported Device Family(1)	UltraScale+™ (GTHE4, GTYE4) Versal® ACAP (GTYE5, GTYP) Zynq® UltraScale+ MPSoC (GTHE4, GTYE4) Zynq UltraScale+ RFSoC
Supported User Interfaces	AXI4-Lite, AXI4-Stream, Native Video, Native SDI
Resources	Performance and Resource Utilization web page
Provided with Subsystem	
Design Files	Hierarchical subsystem packaged with UHD-SDI TX IP core and other IP cores
Example Design	Vivado® IP integrator
Test Bench	N/A
Constraints File	IP cores delivered with XDC files
Simulation Model	N/A
Supported S/W Driver(2)	Standalone and Linux
Tested Design Flows(2)	
Design Entry	Vivado® Design Suite
Simulation	Not Supported
Synthesis	Vivado Synthesis
Support	
Release Notes and Known Issues	Master Answer Record: 68767
All Vivado IP Change Logs	Master Vivado IP Change Logs: 72775
Xilinx Support web page	

Notes:

1. For a complete list of supported devices, see the Vivado IP catalog.
2. Standalone driver details can be found in the Vitis™ directory (<install_directory>/Vitis/<release>/data/embeddedsw/doc/xilinx_drivers.htm). For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Overview

Navigating Content by Design Process

Xilinx® documentation is organized around a set of standard design processes to help you find relevant content for your current development task. This document covers the following design processes:

- **Hardware, IP, and Platform Development:** Creating the PL IP blocks for the hardware platform, creating PL kernels, subsystem functional simulation, and evaluating the Vivado® timing, resource and power closure. Also involves developing the hardware platform for system integration. Topics in this document that apply to this design process include:
  - Port Descriptions
  - High Dynamic Range Data
  - Clocking
  - Resets
  - Customizing and Generating the Subsystem
  - Programming Sequence
  - Example Design

Introduction

The SMPTE UHD-SDI Transmitter (TX) Subsystem allows you to quickly create systems based on SMPTE SDI protocols. It accepts either an AXI4-Stream video stream, native video stream, or native SDI stream. It outputs a native SDI stream when combined with the SMPTE UHD-SDI core. The top level customization parameters select the required hardware blocks needed to build the subsystem. Figure 1-1 shows the subsystem architecture with an AXI4-Stream interface.

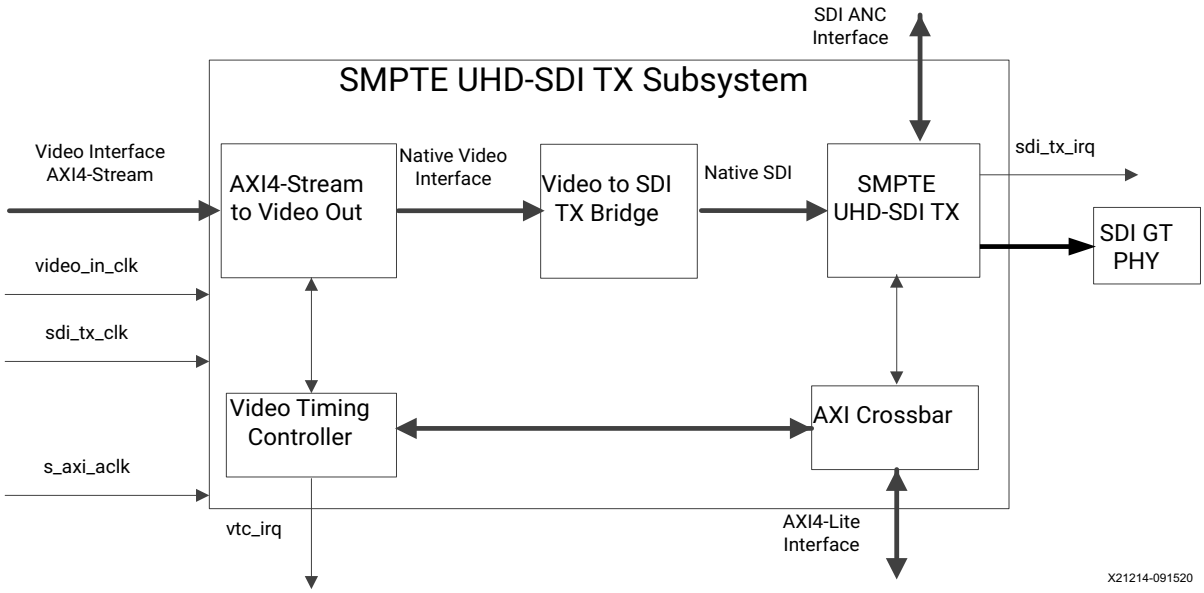

Figure 1-1: **SMPTE UHD-SDI TX Subsystem (AXI4-Stream I/F) Architecture**

Figure 1-2 shows the SMPTE UHD-SDI TX Subsystem with native video as the user interface. The Video Timing Controller (VTC) and AXI4-Stream to Video Out cores are not present in the subsystem and the subsystem accepts native video input.

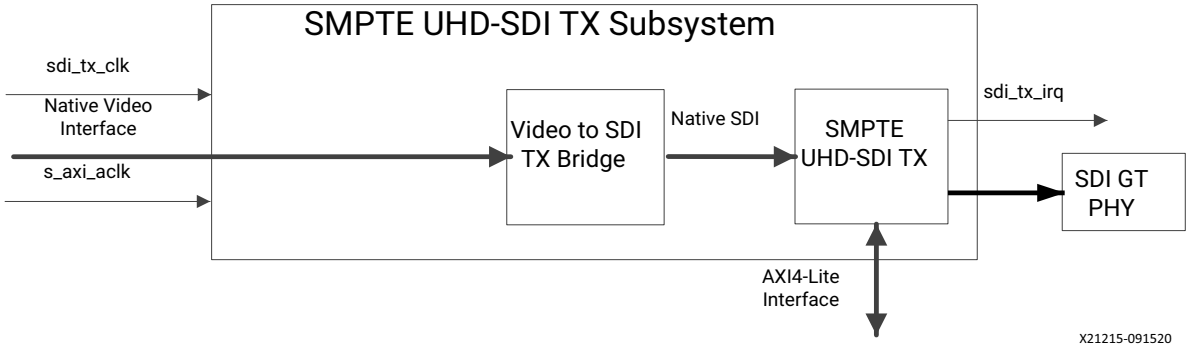

Figure 1-2: **SMPTE UHD-SDI TX Subsystem (Native Video I/F) Architecture**

Figure 1-3 shows the SMPTE UHD-SDI TX Subsystem with native SDI as the user interface. The subsystem accepts native SDI input and it consists only of the SMPTE UHD-SDI TX IP core.

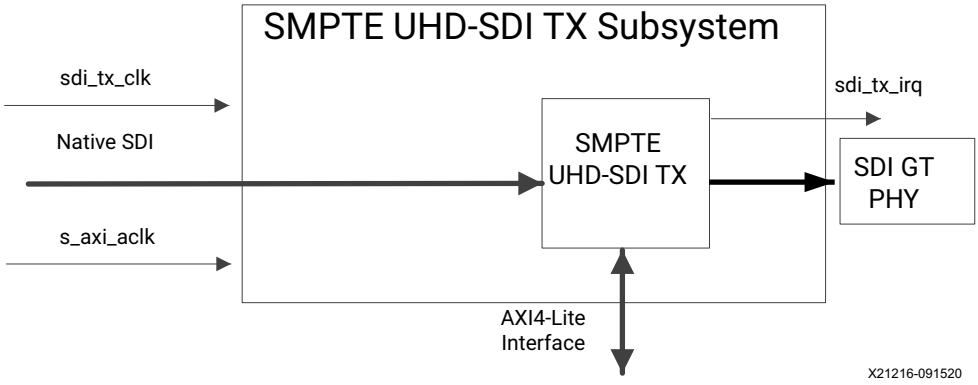

Figure 1-3: **SMPTE UHD-SDI TX Subsystem (Native SDI I/F) Architecture**

The subsystem consists of the following subcores:

- AXI4-Stream to Video Out
- Video to SDI TX Bridge
- SMPTE UHD-SDI TX
- Video Timing Controller
- AXI Crossbar

Subcore Details

AXI4-Stream to Video Out

The AXI4-Stream to Video Out core acts as an interface from the AXI4-Stream interface to a native video interface. This core works with the Xilinx® Video Timing Controller (VTC) core. See the *AXI4-Stream to Video Out Product Guide* (PG044) [Ref 7] for details.

Video to SDI TX Bridge

The LogiCORE IP Video to SDI TX Bridge core converts a native video interface to a native SDI interface. The input is video data with explicit synchronization signals. The output is an SDI virtual interface with one to eight 10-bit data streams and embedded synchronization. Figure 1-4 shows the top level bridge architecture.

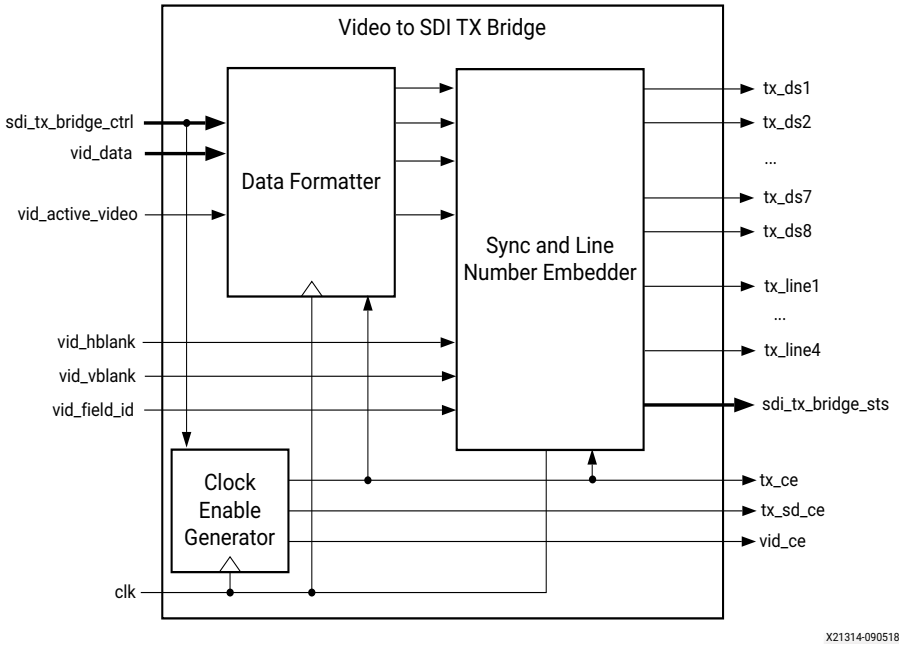

Figure 1-4: **Top-Level Block Diagram of Video to SDI TX Bridge**

The core embeds synchronization packets into the SDI data stream. It creates and embeds line numbers into the SDI data stream. It supports SD-SDI, HD-SDI, 3G-SDI Level A, and 3G-SDI Level B, 6G-SDI and 12G-SDI modes. In addition, it supports YCbCr data format at 10-bit and 12-bit per component (bpc). For SD-SDI and 3G-SDI level B modes, it generates the required clock enables. It automatically re-orders sequential video data to parallel data in 3G Level B. It supports interlaced and progressive line standards.

SMPTE UHD-SDI TX

The SMPTE UHD-SDI TX core receives non-multiplexed native SDI data streams from the SDI TX bridge and generates single multiplexed SDI 10-bit data stream based on the color depth configuration.

Video Timing Controller

The Video Timing controller core is used to generate the Video timing and is used by the AXI4-Stream to Video out core for native video interface signal generation. See the *AXI SmartConnect LogiCORE IP Product Guide* (PG247) [Ref 8] for details.

AXI Crossbar

The AXI Crossbar core is used in the subsystem to route AXI4-Lite requests to corresponding sub-cores based on the address. See the *AXI Interconnect Product Guide* (PG059) [Ref 13] for details.

For general information on the AXI4-Stream interfaces for Xilinx video IP cores, see the *AXI4-Stream Video IP and System Design Guide* (UG934) [Ref 9].

Applications

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

Unsupported Features

The following features are not supported:

- 16-way data stream interleaving is not supported in AXI4-Stream and Native Video interface configurations.
- 3G SDI Level B-Dual stream is not supported in AXI4 Stream and Native Video interface configurations. It is supported only in native SDI configuration.
- YCbCr 4:4:4, YCbCr 4:2:2, and YCbCr 4:2:0 formats are supported. All other formats are not supported in AXI4-stream or Native Video interface configurations.

- SMPTE 2081-10: 2018 only 10-bit 6G/12G HFR resolutions are supported, all other HFR configurations are not supported for AXI4-Stream Interface subsystem configuration.

Licensing and Ordering

The SMPTE UHD-SDI TX Subsystem is provided at no additional cost with the Xilinx Vivado® Design Suite under the terms of the Xilinx End User License.

For more information, visit the UHD-Serial Digital Interface (SDI) product web page.

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.

Note: Data stitching is done by user at the System level. The UHD-SDI TX cores do not provide stitching functionality.

- SMPTE ST 352: Insertion of payload packets into Y Stream and C Stream are supported.

Performance

Maximum Frequencies

The maximum clock frequency for each mode are shown in the following table:

Table 2-1: **Maximum Clock Frequencies**

Mode	TX Clock Maximum Frequency
12G-SDI	297 MHz
6G-SDI, 3G-SDI, and SD-SDI	148.5 MHz
HD-SDI	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 TX Subsystem I/O signals are described in the following tables.

AXI4-Lite Interface Signals

These signals are available when the AXI4-Lite interface option is enabled.

Table 2-2: **AXI4-Lite Interface Signals**

Signal	I/O	Description
s_axi_aclk	I	AXI4-Lite clock
s_axi_arstn	I	AXI4-Lite synchronous reset. Active-Low.
S_AXI_CTRL*		AXI4-Lite interface, defined in the *Vivado Design Suite: AXI Reference Guide* (UG1037) [Ref 11].

Video-Over-AXIS Interface Signals

These signals are available when the Video Interface is set to AXI4-Stream in the Vivado® IDE.

Table 2-3: **Video-Over-AXIS Interface Signals**

Signal	I/O	Description
video_in_clk	I	Video input clock.
video_in_arstn	I	Video input active-Low synchronous reset.

Table 2-3: **Video-Over-AXIS Interface Signals *(Cont'd)***

Signal	I/O	Description
VIDEO_IN_tdata[n-1:0]	O	Video input data for carrying YCbCr 4:4:4 / YCbCr4:2:2 / YCbCr 4:2:0 video with 10 or 12-bit per component, based on the color depth configuration. For 10bpc, n=64 For 12bpc, n=72 (For details see the *AXI4-Stream to Video Out Product Guide* (PG044) [Ref 7] (AXI4-Stream Data Interface Signal Descriptions).
VIDEO_IN_tlast	O	AXI4-Stream TLAST. End of Line.
VIDEO_IN_tready	I	AXI4-Stream TREADY.
VIDEO_IN_tuser	O	AXI4-Stream TUSER. Start of Frame.
VIDEO_IN_tvalid	O	AXI4-Stream TVALID. Active video data enable.
fid	O	Field ID for interlaced videos, fid toggles based on the selected field: 0- even field 1- odd field For progressive videos, fid is set to 0. ***Note:*** fid is expected to toggle in case of psf and 3G-level B progressive videos, as in these modes, one progressive frame is split into two fields. These two fields are transported independently, similar to interlaced video, and again combined and correctly paired together at receiver side with the help of fid signal to construct the original progressive frame. For example, when 1080p 60 Hz video is transported on 3G-SDI level B-DL, the video transport is actually 1080i 60 Hz. The transport is interlaced, but the picture is progressive.

S_AXIS_STS_SB_TX Interface Signals

Table 2-4: **S_AXIS_STS_SB_TX Interface Signals**

Signal	I/O	Description
S_AXIS_STS_SB_TX_tready	O	Core ready
S_AXIS_STS_SB_TX_tvalid	I	Data valid
S_AXIS_STS_SB_TX_tdata[31:0]	I	Sideband signal information from transceiver block Bit 2: gttxresetdone Bit 8: tx_fabric_rst Remaining bit: Unused

M_AXIS_TX Interface Signals

Table 2-5: **M_AXIS_TX Interface Signals**

Signal	I/O	Description
M_AXIS_TX_tready	I	Ready from transceiver block
M_AXIS_TX_tvalid	O	Data valid
M_AXIS_TX_tdata[n-1:0]	O	SDI data output to transceiver block n varies with SDI standard selection. n=40 for 6G-SDI and 12G-SDI n=20 for 3G-SDI
M_AXIS_TX_tuser[31:0]	O	TUSER information - not used.

M_AXIS_CTRL_SB_TX Interface Signals

Table 2-6: **M_AXIS_CTRL_SB_TX Interface Signals**

Signal	I/O	Description
M_AXIS_CTRL_SB_TX__tready	I	Ready from transceiver block
M_AXIS_CTRL_SB_TX_tvalid	O	Data valid
M_AXIS_CTRL_SB_TX_tdata	O	Sideband signal information to transceiver block bit 2-0: tx_mode bit 3: tx_m bit 31-4: 0

Interrupt Signals

Table 2-7: **Interrupt Signals**

Signal	I/O	Description
sdi_tx_irq	O	SMPTE UHD-SDI TX core interrupt. Available only when the Enable AxiLite Interface option is selected.
vtc_irq	O	VTC core interrupt. Available only in AXI4-Stream video interface.

Native SDI Signals

These signals are available when the Video Interface is set to Native SDI in the Vivado IDE. The native SDI also supports an AXI4-Lite control interface. When the AXI4-Lite control interface is enabled, some ports are not available as noted in Table 2-8. This configuration is similar to the transmitter of the SMPTE UHD-SDI IP core in terms of functionality, and

therefore for further information, see the *SMPTE UHD-SDI LogiCORE IP Product Guide* (PG205) [Ref 6].

IMPORTANT: *In Native SDI mode, the image data values must be constrained to avoid SDI prohibited codes. You must avoid the values from 00h to 003h and 3FCh to 3FFh. In Native Video mode and AXI4-Stream mode, the image data values below 004h (000h to 003h) are clamped at 004h, and the values above 3FBh (3FCh to 3FFh) are clamped at 3FBh internally, to avoid the SDI prohibited codes.*

Table 2-8: **Native SDI Signals**

Signal	I/O	Description
sdi_tx_clk	I	SMPTE UHD-SDI TX core clock
sdi_tx_rst	I	Active-High reset
sdi_tx_ctrl[31:0](2)	I	Bit0: module_enable Bit1: not used Bit3–bit2: reserved Bit6–bit4: tx_mode:(1) 000-HD 001-SD 010-3G 100-6G 101-12G; Bit7: tx_m (tx_rate): 0 – integer frame rate 1 – fractional frame rate (frame_rate/1.001) Bit10–bit8: tx_mux_pattern: 000-SD, HD, and 3G level A 001-3G level B 010-8 stream interleave in 6G and 12G modes 011-4 stream interleave in 6G mode Bit11: reserved Bit12: tx_insert_crc Bit13: tx_insert_st352 Bit14: tx_overwrite_st352 Bit15: tx_st352_f2_en Bit16: tx_insert_sync_bit Bit17: tx_sd_bitrep_bypass Bit18: tx_use_anc_in Bit19: tx_insert_ln Bit20: tx_insert_edh Bit23: tx_insert_c_str_st352 Bit24: st352_str_switch_3g_a Bit31–bit25: reserved
ST352_DATA_IN_tx_st352_data_ch0[31:0](2)	I	ST352 data for Y stream of channel 0
ST352_DATA_IN_tx_st352_data_ch1[31:0](2)	I	ST352 data for Y stream of channel 1
ST352_DATA_IN_tx_st352_data_ch2[31:0](2)	I	ST352 data for Y stream of channel 2
ST352_DATA_IN_tx_st352_data_ch3[31:0](2)	I	ST352 data for Y stream of channel 3
ST352_DATA_IN_tx_st352_data_ch4[31:0](2)	I	ST352 data for Y stream of channel 4

Table 2-8: **Native SDI Signals *(Cont'd)***

Signal	I/O	Description
ST352_DATA_IN_tx_st352_data_ch5[31:0](2)	I	ST352 data for Y stream of channel 5
ST352_DATA_IN_tx_st352_data_ch6[31:0](2)	I	ST352 data for Y stream of channel 6
ST352_DATA_IN_tx_st352_data_ch7[31:0](2)	I	ST352 data for Y stream of channel 7
ST352_DATA_IN_C_tx_st352_data_ch0[31:0](2)(4)	I	ST352 data for C stream of channel 0
ST352_DATA_IN_C_tx_st352_data_ch1[31:0](2)(4)	I	ST352 data for C stream of channel 1
ST352_DATA_IN_C_tx_st352_data_ch2[31:0](2)(4)	I	ST352 data for C stream of channel 2
ST352_DATA_IN_C_tx_st352_data_ch3[31:0](2)(4)	I	ST352 data for C stream of channel 3
ST352_DATA_IN_C_tx_st352_data_ch4[31:0](2)(4)	I	ST352 data for C stream of channel 4
ST352_DATA_IN_C_tx_st352_data_ch5[31:0](2)(4)	I	ST352 data for C stream of channel 5
ST352_DATA_IN_C_tx_st352_data_ch6[31:0](2)(4)	I	ST352 data for C stream of channel 6
ST352_DATA_IN_C_tx_st352_data_ch7[31:0](2)(4)	I	ST352 data for C stream of channel 7
ST352_DATA_IN_tx_st352_line_f1	I	Odd line to insert ST352 data. This is used for progressive video and you have to control the tx_st352_f2_en bit of the sdi_tx_ctrl bus.(1)
ST352_DATA_IN_tx_st352_line_f2	I	Even line to insert ST352 data
SDI_DS_IN_ds1[9:0]	I	SDI data stream 1
SDI_DS_IN_ds2[9:0]	I	SDI data stream 2
SDI_DS_IN_ds3[9:0]	I	SDI data stream 3
SDI_DS_IN_ds4[9:0]	I	SDI data stream 4
SDI_DS_IN_ds5[9:0]	I	SDI data stream 5
SDI_DS_IN_ds6[9:0]	I	SDI data stream 6
SDI_DS_IN_ds7[9:0]	I	SDI data stream 7
SDI_DS_IN_ds8[9:0]	I	SDI data stream 8
SDI_DS_IN_ds9[9:0] to SDI_DS_IN_ds16[9:0](4)	I	SDI data stream 9 to 16
SDI_DS_IN_ln_num_1[10:0] to SDI_DS_IN_ln_num_4[10:0]	I	SDI data stream line number 1 to 4
SDI_DS_IN_ln_num_5[10:0] to SDI_DS_IN_ln_num_8[10:0](4)	I	SDI data stream line number 5 to 8
SDI_DS_IN_tx_ce	I	Clock enable
SDI_DS_IN_tx_sd_ce	I	SD-SDI mode clock enable
sdi_tx_err[31:0](2)	O	Bit0: tx_ce_align_err Bit31 to bit1: Reserved

Notes:

1. See the TX Ports section of the *SMPTE UHD-SDI LogiCORE IP Product Guide* (PG205) [Ref 6] for more details on the signal descriptions.
2. Enabled only when the Enable AxiLite interface option is *not* selected in the Vivado IDE.
3. Enabled only when the Insert ST352 in C stream option is selected in the Vivado IDE.
4. Enabled only when Native SDI is selected as the video interface *and* 12G SDI 16DS is selected as the SDI standard.

Native Video (VID_IO_IN) Interface Signals

These signals are available when the Video Interface is set to Native Video in the Vivado IDE.

Table 2-9: **VID_IO_IN Interface Signals**

Signal	I/O	Description
VID_IO_IN_data[n-1:0]	I	Native video input data Where n=bpc*ppc*3. For 10bpc at 2ppc, n=10*2*3=60 For 12bpc at 2ppc, n=12*2*3=72
VID_IO_IN_active_video	I	Native video active video input
VID_IO_IN_field	I	Native video field input For interlaced videos, fid toggles based on the field selected: 0- even field 1- odd field For progressive videos, fid is set to 0. ***Note:*** fid is expected to toggle in case of psf and 3G-level B progressive videos, as in these modes, one progressive frame is split into two fields. These two fields are transported independently, similar to interlaced video, and again combined and correctly paired together at receiver side with the help of fid signal to construct the original progressive frame. For example, when 1080p 60 Hz video is transported on 3G-SDI level B-DL, the video transport is actually 1080i 60 Hz. The transport is interlaced, but the picture is progressive.
VID_IO_IN_hblank	I	Native video hblank input
VID_IO_IN_vblank	I	Native video vblank input
vid_ce	O	Native video clock enable

The native video interface is expected to start from the start of active video or start of blanking. In interlaced video format, the data of field 0 comes before field 1.

Vertical timing is defined by vblank signal. VBLANK is 1 indicates, vertical blanking lines. Similarly, for horizontal timing information, hblank signal is used. Active video is sent only

when hblank and vblank is 0 and `active_video` signal is asserted. A sample of native video interface timing is shown in the following image.

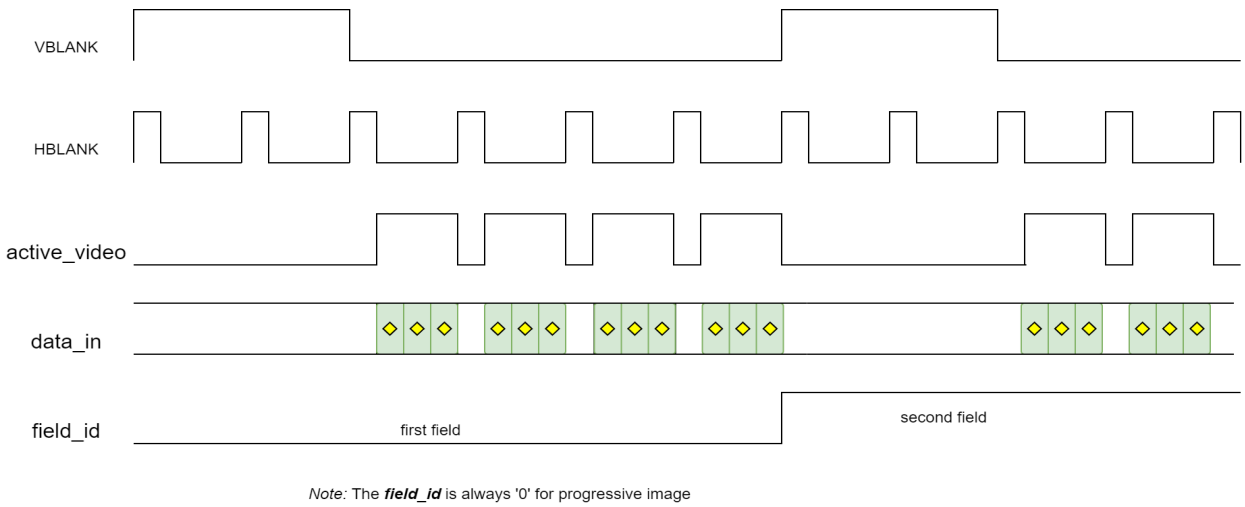

Figure 2-1: **Native Video Interface Timing**

SDI_TX_ANC_DS_OUT Interface Signals

These signals are available when the Ancillary Data (ANC) Insertion I/F option is selected.

Table 2-10: **SDI_TX_ANC_DS_OUT Interface Signals**

Signal	I/O	Description
SDI_TX_ANC_DS_OUT_tx_ds1_st352_out[9:0] to SDI_TX_ANC_DS_OUT_tx_ds8_st352_out[9:0]	O	SDI stream outputs from 1 to 8 after ST352 insertion
SDI_TX_ANC_DS_OUT_tx_ds9_st352_out[9:0] to SDI_TX_ANC_DS_OUT_tx_ds16_st352_out[9:0]	O	SDI stream outputs from 9 to 16 after ST352 insertion. Enabled only when native SDI and 12G SDI 16 DS option is selected.

SDI_TX_ANC_DS_IN Interface Signals

These signals are available when the ANC Data Insertion I/F option is selected.

Table 2-11: **SDI_TX_ANC_DS_IN Interface Signals**

Signal	I/O	Description
SDI_TX_ANC_DS_IN_tx_ds1_anc_in[9:0] to SDI_TX_ANC_DS_IN_tx_ds8_anc_in [9:0]	I	SDI data stream from 1 to 8 after ancillary data insertion.

Table 2-11: **SDI_TX_ANC_DS_IN Interface Signals *(Cont'd)***

Signal	I/O	Description
SDI_TX_ANC_DS_IN_tx_ds9_anc_in[9:0] to SDI_TX_ANC_DS_IN_tx_ds16_anc_in [9:0]	I	SDI data stream from 9 to 16 after ancillary data insertion. Enabled only when native SDI and12 SDI 16 DS option is selected.
sdi_tx_anc_ctrl_out[31:0]	O	Bit0: tx_sd_ce Bit1: tx_ce Bit12–Bit2: SDI line number Bit13: SID TX IP internal reset Bit16–Bit14: tx_mode Bit19–Bit17: tx_mux_pattern Bit31–Bit20: reserved

High Dynamic Range Data

This SMPTE UHD-SDI TX IP supports only HLG HDR.

When AXI4-Lite registers are enabled (C_INCLUDE_AXILITE is true), user needs to program ST352 payload accordingly, through registers allocated. As per ST 2082-10_2018 specification, bits 4 and 5 of byte 2 in the payload are used to configure HLG HDR.

When AXI4-Lite registers are disabled (C_INCLUDE_AXILITE is false), use the Native SDI interface ports for sending ST352 payload information. Refer Table 2-8 for the ST352 related ports

Register Space

This section details registers available in the SMPTE UHD-SDI TX Subsystem. The address map is split into following regions:

- SMPTE UHD-SDI TX core
- Video Timing Controller (VTC) core

The VTC core is enabled only for the AXI4-Stream interface configuration. Each IP core is given an address space of 64K. Example offset addresses from the system base address when the SMPTE UHD-SDI TX and VTC core registers are enabled are shown in Table 2-12.

Table 2-12: **Subcore Address Offsets**

IP Core	Offset
SMPTE UHD-SDI TX	0x0_0000
VTC	0x1_0000

SMPTE UHD-SDI TX Registers

The SMPTE UHD-SDI TX registers are available when **Enable AXI4-Lite Interface** is selected in the Vivado IDE. The SMPTE UHD-SDI TX IP core register space is shown in Table 2-13.

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

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

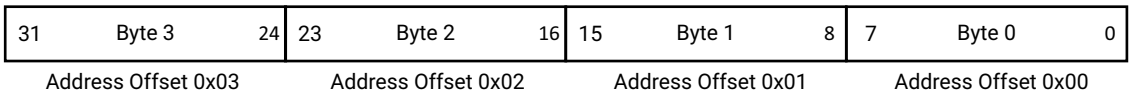

Figure 2-2: **32-bit Little Endian Example**

Table 2-13: **SMPTE UHD-SDI TX 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	TX_ST352_LINE	32-bit	R/W	ST352 packet insertion line number
0x1C	TX_ST352_DATA_DS1	32-bit	R/W	Data stream 1 ST352 packet data
0x20	TX_ST352_DATA_DS3	32-bit	R/W	Data stream 3 ST352 packet data
0x24	TX_ST352_DATA_DS5	32-bit	R/W	Data stream 5 ST352 packet data
0x28	TX_ST352_DATA_DS7	32-bit	R/W	Data stream 7 ST352 packet data
0x2C	TX_ST352_DATA_DS9	32-bit	R/W	Data stream 9 ST352 packet data
0x30	TX_ST352_DATA_DS11	32-bit	R/W	Data stream 11 ST352 packet data
0x34	TX_ST352_DATA_DS13	32-bit	R/W	Data stream 13 ST352 packet data
0x38	TX_ST352_DATA_DS15	32-bit	R/W	Data stream 15 ST352 packet data
0x3C	VERSION	32-bit	RO	Version Register
0x40	SS_CONFIG	32-bit	RO	IP core Configuration
0x68	SDI_TX_BRIDGE_STS	32-bit	RO	SDI TX Bridge Status
0x6C	AXI4S_VID_OUT_STS	32-bit	R/W	AXI4-Stream Video Out status register
0x70	TX_ST352_DATA_DS2	32-bit	R/W	Data stream 2 ST352 packet data
0x74	TX_ST352_DATA_DS4	32-bit	R/W	Data stream 4 ST352 packet data
0x78	TX_ST352_DATA_DS6	32-bit	R/W	Data stream 6 ST352 packet data
0x7C	TX_ST352_DATA_DS8	32-bit	R/W	Data stream 8 ST352 packet data

Table 2-13: **SMPTE UHD-SDI TX IP Core Register Space *(Cont'd)***

Offset	Name	Width	Access	Description
0x80	TX_ST352_DATA_DS10	32-bit	R/W	Data stream 10 ST352 packet data
0x84	TX_ST352_DATA_DS12	32-bit	R/W	Data stream 12 ST352 packet data
0x88	TX_ST352_DATA_DS14	32-bit	R/W	Data stream 14 ST352 packet data
0x8C	TX_ST352_DATA_DS16	32-bit	R/W	Data stream 16 ST352 packet data

Notes:

1. Access type and reset value for all the reserved bit 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 (6:0) of the read and write address of the AXI4-Lite interface are decoded. This means that accessing addresses 0x00 and 0x80 results in reading the same address as 0x00.
4. Reads and writes to addresses outside this table do not return an error.

RST_CTRL Register (0x00)

The core control register allows you to enable and disable the SMPTE UHD-SDI TX IP core and apply a soft reset during core operation.

Table 2-14: **RST_CTRL Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:10	Reserved	RO	0	Reserved
9	AXI4S_VID_OUT_EN	R/W	0	Enable bits for AXI4-Stream to Video out core 1 – AXI4-Stream to Video out core is enabled 0 – AXI4-Stream to Video out core is disabled. This bit is enabled only for the AXI4-Stream interface and has no impact on native video and native SDI interface subsystem configurations.
8	SDITX_BRIDGE_EN	R/W	0	Enable bits for SDI TX Bridge 1 – SDI TX Bridge is enabled 0 – SDI TX Bridge is disabled This bit is not enabled for the native SDI interface subsystem.
7:2	Reserved	RO	0	Reserved
1	SRST	R/W	0	Soft reset for SDI TX IP core If 1 is written to this bit, all registers of the SDI TX IP core are reset.
0	SDITX_IP_EN	R/W	0	Enable bits for SDI TX IP core 1 – SDI TX IP core is enabled 0 – SDI TX IP core is disabled

MODULE_CTRL Register (0x04)

The module control register allows you to control the SMPTE UHD-SDI TX IP core and to change the IP core functional modes.

Table 2-15: **MODULE_CTRL Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:25	Reserved	RO	0	Reserved
24	TX_ST352_STR_SWITCH_3G_A	R/W	0	When this is set to 1'b1, the ST352 value from the TX_ST352_DATA_DS2 register (0x70 offset) is used instead of TX_ST352_DATA_DS3 (0x20 offset)
23	TX_INSERT_C_STR_ST352	R/W	0	This bit controls whether ST352 has to be inserted into the channel C stream. Contents of registers from offset, 0x70 to 0x8C are used for the ST352 payload
22:21	FMT_SEL	R/W	0	YCbCr444, YCbCr422, or YCbCr420 color space selection 2'b00 : YCbCr422 format 2'b01 : YCbCr420 format 2'b10 : YCbCr444 format This field is not enabled for Native SDI interface.
20	TX_INSERT_EDH	R/W	0	When this bit is High, the transmitter generates and inserts EDH packets into every field in SD-SDI mode. When this bit is Low, EDH packets are not inserted. This bit is ignored in all modes except SD-SDI mode.
19	TX_INSERT_LN	R/W	0	When this bit is High, the transmitter inserts line numbers into all active data streams after the EAV of each video line. The line numbers must be supplied on the tx_line_ch*n* input ports of all active data stream pairs. When this bit is Low, line numbers are not inserted. This bit is ignored in SD-SDI mode.
18	TX_USE_ANC_IN	R/W	0	When Low, the data streams out of the ST352 packet insertion function are routed internally to the TX output channels. When High, the TX output channels accept data streams from the tx_ds[16:1]_anc_in ports.
17	TX_SD_BITREP_BYPASS	R/W	0	This bit bypasses the 11 times bit replicator used in SD-SDI mode when High. For normal operation with Xilinx serial transceiver transmitters, this input must be Low so that the bit replicator function is active.

Table 2-15: **MODULE_CTRL Register Bit Mapping *(Cont'd)***

Bits	Name	Access	Default Value	Description
16	TX_INSERT_SYNC_BIT	R/W	0	In 6G and 12G modes, when this bit is High, the sync bit insertion function is enabled for run length mitigation.
15	TX_ST352_F2_EN	R/W	0	This bit controls whether or not ST 352 packets are inserted on the line indicated by tx_vpid_line_f2
14	TX_OVERWRITE_ST352	R/W	0	If this bit is High, ST 352 packets already present in the data streams are overwritten. If this bit is Low, existing ST 352 packets are not overwritten.
13	TX_INSERT_ST352	R/W	0	When this bit is High, ST 352 packets are inserted into the data streams, otherwise the ST352 packets are not inserted.
12	TX_INSERT_CRC	R/W	0	When this bit is High, the transmitter generates and inserts CRC values into the data streams for each video line in all modes except SD-SDI. When this bit is Low, CRC values are not inserted into the data streams. This bit is ignored in SD-SDI mode.
11	Reserved	RO	0	Reserved
10:8	TX_MUX_SEL	R/W	0	Internal TX mux pattern which specifies the data stream interleaving pattern to be used: 3'b000 : SD-SDI,HD-SDI,and 3G-SDI level A 3'b001 : 3G-SDI level B 3'b010 : 8 stream interleave in 6G-SDI and 12G-SDI modes 3'b011 : 4 stream interleave in 6G-SDI mode 3'bs100-16 stream interleave in 12G-SDI mode
7	TX_M	R/W	0	0 – integer frame rate 1 – fractional frame rate (frame_rate/1.001)
6:4	SDITX_SS_MODE	R/W	0	TX Mode 3'b000 : HD-SDI mode 3'b001 : SD-SDI mode 3'b010 : 3G-SDI mode Level A Mode if SDI TX bridge is enabled; 3G-SDI mode if SDI TX bridge is not enabled 3'b011: 3G-SDI Level B Mode if SDI TX bridge is enabled; NA if SDI TX bridge is not enabled 3'b100 : 6G-SDI mode 3'b101 : 12G-SDI mode
3	ENABLE_HFR	R/W	0	When this bit is 1,the resolutions transmitted is HFR
2:0	Reserved	RO	0	Reserved

Global Interrupt Enable Register (GLBL_IER) (0x0C)

Table 2-16: **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 bit

Interrupt Status Register (ISR) (0x10)

The interrupt status register captures the error and status information for the IP core.

Table 2-17: **ISR Bit Mapping**

Bits	Name	Access[(1)]	Default Value	Description
31:11	Reserved	RO	0	Reserved
10	UNDERFLOW_INTR	R/W1C	0	AXI4-Stream to Video out core underflow indication. This bit is enabled for the AXI4-Stream interface.
9	OVERFLOW_INTR	R/W1C	0	AXI4-Stream to Video out core overflow indication. This bit is enabled for the AXI4-Stream interface.
8	AXI4S_VID_LOCK_INTR	R/W1C	0	Lock indication from AXI4-Stream to Video out core. This bit is enabled for the AXI4-Stream interface.
7:3	Reserved	RO	0	Reserved
2	VSYNC_VALID_INTR	R/W1C	0	Asserted when Video sync has been detected at the start of each frame.
1	TX_CE_ALIGN_ERR_INTR	R/W1C	0	This bit indicates problems with the 5/6/5/6 clock cycle cadence of the tx_sd_ce input in SD-SDI mode. In SD-SDI mode, the tx_sd_ce signal must follow a regular 5/6/5/6 clock cycle cadence. If it does not, the SD-SDI serial stream is formed incorrectly. The TX_CE_ALIGN_ERR_INTR bit goes High if the cadence is incorrect.
0	GTTX_RSTDONE_INTR	R/W1C	0	Asserted when GTTX_RESETDONE is High

Notes:
1. W1C – Write 1 to Clear (to clear register bit, write 1 to the corresponding bit).

Interrupt Enable Register (IER) (0x14)

The interrupt enable register 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 2-18: **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 2-17.
10	UNDERFLOW_IE	R/W	0	
9	OVERFLOW_IE	R/W	0	
8	AXI4S_VID_LOCK_IE	R/W	0	
7:3	Reserved	RO	0	
2	VSYNC_VALID_IE	R/W	0	
1	TX_CE_ALIGN_ERR_IE	R/W	0	
0	GTTX_RSTDONE_IE	R/W	0	

TX_ST352_LINE Register (0x18)

Table 2-19: **TX_ST352_LINE Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:27	Reserved	RO	0	Reserved
26:16	TX_ST352_F2_LN	R/W	0	Line number used to insert st352 packet for field 2
15:11	Reserved	RO	0	Reserved
10:0	TX_ST352_F1_LN	R/W	0	Line number used to insert st352 packet for field 1

TX_ST352_DATA_DS1 Register (0x1C)

Table 2-20: **TX_ST352_DATA_DS1 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS1	R/W	0	ST 352 payload ID packet data bytes captured from data stream 1

TX_ST352_DATA_DS3 Register (0x20)

Table 2-21: **TX_ST352_DATA_DS3 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS3	R/W	0	ST 352 payload ID packet data bytes captured from data stream 3

TX_ST352_DATA_DS5 Register (0x24)

Table 2-22: **TX_ST352_DATA_DS5 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS5	R/W	0	ST 352 payload ID packet data bytes captured from data stream 5

TX_ST352_DATA_DS7 Register (0x28)

Table 2-23: **TX_ST352_DATA_DS7 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS7	R/W	0	ST 352 payload ID packet data bytes captured from data stream 7

TX_ST352_DATA_DS9 Register (0x2C)

Table 2-24: **TX_ST352_DATA_DS9 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS9	R/W	0	ST 352 payload ID packet data bytes captured from data stream 9

TX_ST352_DATA_DS11 Register (0x30)

Table 2-25: **TX_ST352_DATA_DS11 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS11	R/W	0	ST 352 payload ID packet data bytes captured from data stream 11

TX_ST352_DATA_DS13 Register (0x34)

Table 2-26: **TX_ST352_DATA_DS13 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS13	R/W	0	ST 352 payload ID packet data bytes captured from data stream 13

TX_ST352_DATA_DS15 Register (0x38)

Table 2-27: **TX_ST352_DATA_DS15 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS15	R/W	0	ST 352 payload ID packet data bytes captured from data stream 15

VERSION Register (0x3C)

Table 2-28: **VERSION Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	VERSION	RO	32'h02_00_0_0_00	For uhd_sdi_tx_ss_v1_0, the VERSION REGISTER is 32'h01_00_0_0_00. For uhd_sdi_tx_ss_v2_0, the VERSION REGISTER is 32'h02_00_0_0_00. • [31:24] - Subsystem major version. • [23:16] - Subsystem minor version. • [15:12] - Subsystem version revision. • [11:8] - Subsystem Patch details. • [7:0] - Internal revision.

SS_CONFIG Register (0x40)

Table 2-29: **SS_CONFIG Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:6	Reserved	RO	0	Reserved
5	TX_INSERT_C_STR_ST352	RO	0	Set if the subsystem is generated with Insert ST352 in C stream
4	ANC_IF	RO	0	ANC Interface Enable
3:2	VID_INTF	RO	0	Video Interface 2'b00 : AXI4-Stream interface 2'b01 : Native video interface 2'b10 : Native SDI interface
1	INC_TX_EDH_PROC	RO	1	Set if the IP core is generated with INCLUDE_TX_EDH_PROCESSOR
0	Reserved	RO	0	Reserved

SDI_TX_BRIDGE_STS Register (0x68)

This register is not updated when the native SDI interface is selected.

Table 2-30: **SDI_TX_BRIDGE_STS Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:7	Reserved	RO	0	Reserved
6	BRIDGE_3G_LEVEL_B	RO	0	Asserted High when incoming stream is 3G-SDI level B
5:4	3GBRIDGE_TX_MODE	RO	0	3G Bridge TX mode 2'b00 : HD-SDI mode 2'b01 : SD-SDI mode 2'b10 : 3G-SDI mode
3:1	Reserved	RO	0	Reserved
0	SDITX_BRIDGE_SEL	RO	0	Select bits for SDI TX Bridge 0 – 3G SDI TX Bridge is selected 1 – 12G SDI TX Bridge is selected

AXI4S_VID_OUT_STS Register (0x6C)

This register is available only when AXI4-Stream is selected.

Table 2-31: **AXI4S_VID_OUT_STS Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	AXI4S_VID_OUT_STS	RO	0	Status[31:0] from AXI4-Stream to Video Out core

TX_ST352_DATA_DS2 Register (0x70)

Table 2-32: **TX_ST352_DATA_DS2 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS2	R/W	0	ST 352 payload ID packet data bytes captured from data stream 2

TX_ST352_DATA_DS4 Register (0x74)

Table 2-33: **TX_ST352_DATA_DS4 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS4	R/W	0	ST 352 payload ID packet data bytes captured from data stream 4

TX_ST352_DATA_DS6 Register (0x78)

Table 2-34: **TX_ST352_DATA_DS6 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS6	R/W	0	ST 352 payload ID packet data bytes captured from data stream 6

TX_ST352_DATA_DS8 Register (0x7C)

Table 2-35: **TX_ST352_DATA_DS8 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS8	R/W	0	ST 352 payload ID packet data bytes captured from data stream 8

TX_ST352_DATA_DS10 Register (0x80)

Table 2-36: **TX_ST352_DATA_DS10 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS10	R/W	0	ST 352 payload ID packet data bytes captured from data stream 10

TX_ST352_DATA_DS12 Register (0x84)

Table 2-37: **TX_ST352_DATA_DS12 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS12	R/W	0	ST 352 payload ID packet data bytes captured from data stream 12

TX_ST352_DATA_DS14 Register (0x88)

Table 2-38: **TX_ST352_DATA_DS14 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS14	R/W	0	ST 352 payload ID packet data bytes captured from data stream 14

TX_ST352_DATA_DS16 Register (0x8C)

Table 2-39: **TX_ST352_DATA_DS16 Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:0	TX_ST352_DATA_DS16	R/W	0	ST 352 payload ID packet data bytes captured from data stream 16

VTC Registers

The VTC registers are available when **AXI4-Stream** is selected for the video interface in the Vivado IDE. For details about VTC registers, see the *Video Timing Controller Product Guide* [Ref 8] for details.

Designing with the Subsystem

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

General Design Guidelines

This section describes the steps required to turn a SMPTE UHD-SDI TX 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 TX Subsystem that is created is delivered with an example design that can be implemented in Xilinx® devices. 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

The SMPTE UHD-SDI TX 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 TX 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 TX 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 TX Subsystem can only be made by selecting options from the Vivado® Integrated Design Environment (IDE).

Clock Frequency Selection

The SMPTE UHD-SDI TX 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. The frequency range is 50 MHz to 150 MHz.
sdi_tx_clk	Core clock for the SMPTE UHD-SDI TX core. See Table 3-2 for more details.

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

Clock Name	Description
video_in_clk	Available only when the AXI4-Stream interface is selected. Clock used for video data conversion to SDI data stream. To support 12G-SDI for 10-bit YCBCr 4:2:2 in 2 Pixel Per Clock (PPC)(2), the clock must be set to a maximum of 300 MHz. 2*(BPC)(1)*(PPC)*clock = 2*10*2*300 MHz = 12 Gb/s To support 12G-SDI for 12-bit YCBCr 4:2:2 in 2 PPC, the clock must be set to a maximum of 250 MHz. 2*(BPC)*(PPC)*clock=2*12*2*250 = 12 G The video_in_clk for the SMPTE UHD-SDI TX Subsystem must not be less than sdi_tx_clk (causes underflow). Use caution on the overflow if you are using a value much higher than sdi_tx_clk.

Notes:

1. BPC is set to either 10 or 12 because the subsystem supports 10-bit or 12-bit YCBCr4:2:2.
2. PPC is set to 2 by the SDI bridge.

The frequency of the SMPTE UHD-SDI TX core clock, `sdi_tx_clk`, is shown in Table 3-2.

Table 3-2: **SMPTE UHD-SDI TX Clock**

SMPTE Standard	Supported Data Stream	Clock Frequency (MHz)
SD-SDI	1	148.5 (27 MHz sampling at tx_sd_ce with 5-6-5-6 cadence)
HD-SDI	2	74.25
3G-SDI Level A	2	148.5 ***Note:*** tx_sd_ce is used with 50 duty cycle for 3G Level A/Level B
3G-SDI Level B	4	
6G-SDI	8	148.5
12G-SDI	8	297

See the Clocking section in the *SMPTE UHD-SDI LogiCORE IP Product Guide* (PG205) [Ref 6] for more information.

CPLL Clocking

Note: For Versal® ACAP GT clocking information, refer to the *Versal ACAP GTY Transceivers Architecture Manual* [Ref 18].

The following figure shows the UltraScale GT clocking architecture.

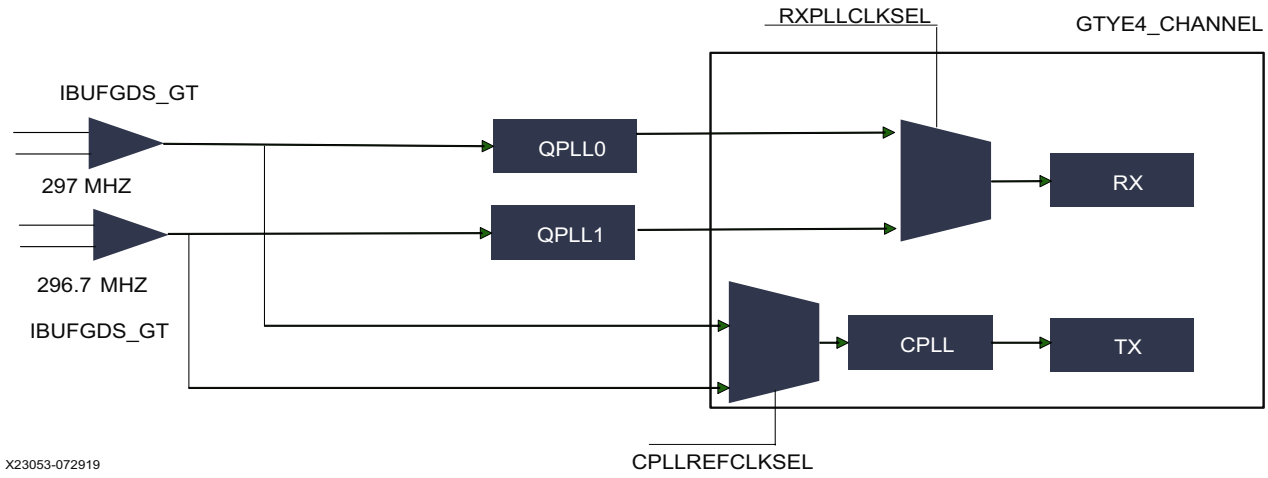

Figure 3-1: **GT Clocking Architecture in Loopback Example Design**

Two reference clocks are used to support integer and fractional line rates of SDI. The reference clock for QPLL0 is fixed to 297 MHz from the on-board Si570 chip. The reference clock for the CPLL is fixed at 296.7 MHz from Si5328 chip output. The CPLL switches between the 297 MHz and the 296.7 MHz reference clocks using the CPLLREFCLKSEL.

IMPORTANT: *When using QPLL0 and QPLL1 for 12G-SDI integer and fractional (1/1.001) rate change, switching between rates on the SDI-RX can introduce a glitch on the clock which in turn introduces CRC errors on the TX channel. CRC errors do not occur in SD-SDI/HD-SDI/3-G SDI/6-G SDI integer/fractional modes with QPLL0 and QPLL1 clocking combination. For more information, see Answer Records 72254 and 72449. Therefore, it is not recommended to use this clocking configuration when both transmit and receive 12G-SDI integer and fractional modes use the same transceiver. If required, Xilinx recommends to use a CPLL-QPLL combination with CPLL for TX and QPLL0/1 for RX as shown in Figure 3-1.*

The integer and fractional rates for TX can be selected using a CPLL reference clock input selection with 297 MHz and 296.7 MHz respectively. This CPLL/QPLL clocking combination is not feasible with -1 speed grade devices because CPLL does not support 12G-SDI line rates. You need to select an UltraScale+™ GTH/GTY -2 speed grade or faster rate with >0.85V. Refer to the respective FPGA device data sheets for CPLL line rate limits. The UHD-SDI example designs are updated to use the CPLL and QPLL clocking combination. The UHD-SDI GT IP is updated to provide CPLL support from version 2019.2 or later.

Note: For more information refer to PG290 (KCU116 Example design section) [Ref 10].

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_in_arstn: Active-Low reset for the subsystem blocks and synchronous with `video_in_clk`.
- sdi_tx_rst: Active-High reset for the SMPTE UHD-SDI TX core and synchronous with `sdi_tx_clk`. See the *Clocking* section the *SMPTE UHD-SDI Product Guide* [Ref 6].

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

Table 3-3: **Core Resets**

Sub-Core	s_axi_arstn	video_in_arstn	sdi_tx_rst
AXI4-Stream to Video Out	N/A	N/A	NA
Video to SDI TX Bridge	N/A	N/A	Connected to rst core port
SMPTE UHD-SDI TX	Connected to s_axi_aresetn core port	Connected to axis_rstn core port	Connected to tx_rst core port
Video Timing Controller	Connected to s_axi_aresetn core port	N/A	N/A
AXI Crossbar	Connected to aresetn core port	N/A	N/A

Note: The effect of each reset (`s_axi_arstn`, `video_in_arstn`, `sdi_tx_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. All the resets should be active until the associated clocks are stable.

UHD-SDI Audio Embed Use Case

The SMPTE UHD-SDI TX Subsystem provides an ancillary data (ANC) interface controlled by the Enable ANC Data (Incl. Audio) Insertion I/F parameter in the Vivado® IDE and can be used to embed the SDI audio in the SDI video stream. This section provides an overview of embedding SDI audio into native SDI stream(s) using the Xilinx® UHD-SDI Audio IP core. See the *UHD-SDI Audio LogiCORE IP Product Guide* (PG309) [Ref 14] for more details. The following figure shows the UHD-SDI audio embed use case.

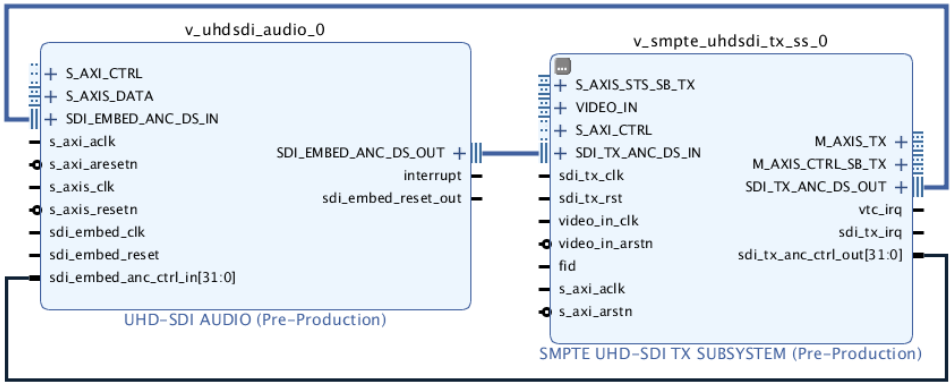

Figure 3-2: **UHD-SDI Audio Embed Use Case**

The Xilinx® UHD-SDI Audio IP core is configurable as an audio embedder or an audio extractor. When configured as an audio embedder, it can embed up to 32 channels (16 channel pairs) of audio onto an SDI stream.

The UHD-SDI Audio IP is designed in accordance with SMPTE ST 272 for SD-SDI and SMPTE ST 299 for HD/3G/6G/12G SDI. It supports audio embedding at multiple audio sample rates (32 kHz, 44.1 kHz and 48 kHz).

Functional Description

The SDI interface to the audio embedder consists of input and output ancillary data streams from the SMPTE UHD-SDI TX Subsystem. The audio interface is a 32-bit AXI4-Stream slave bus. The following figure shows the UHD-SDI Audio (Embed) IP block diagram.

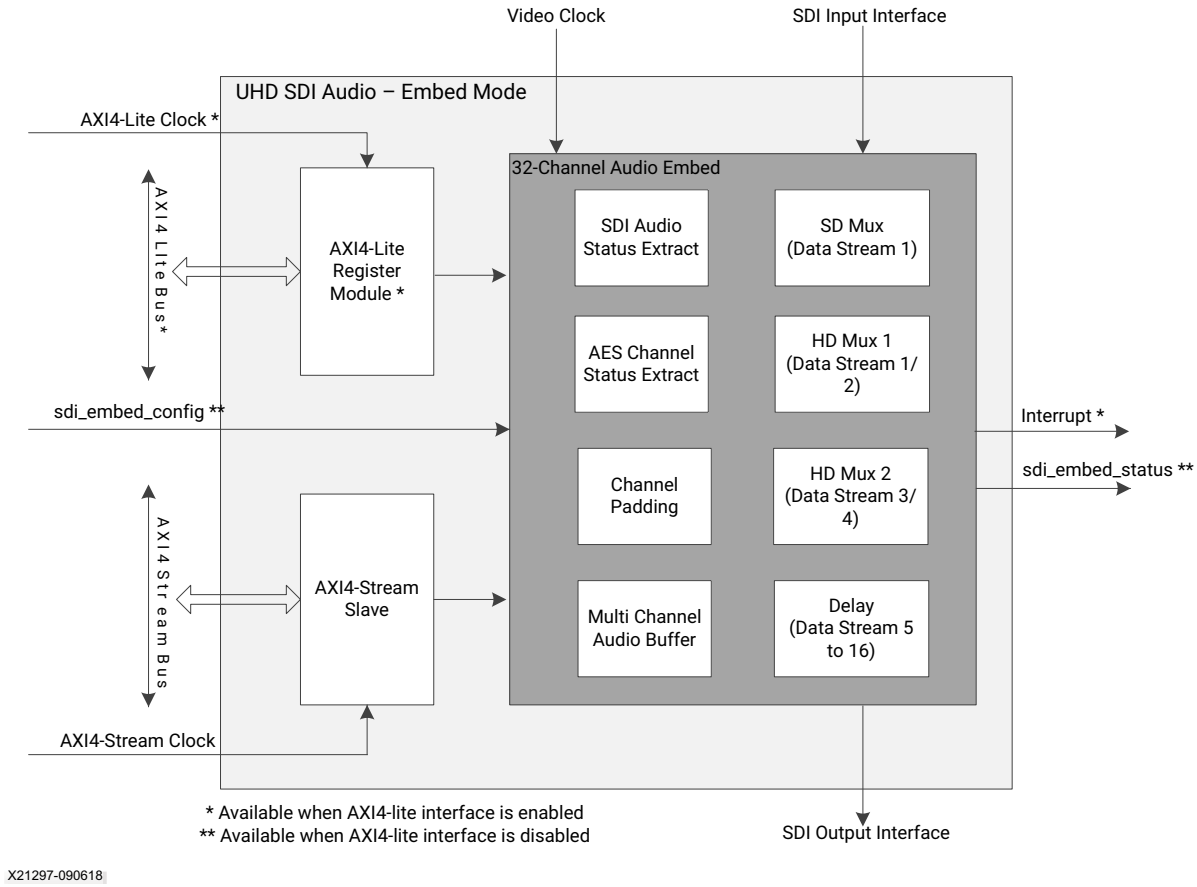

Figure 3-3: **UHD-SDI Audio (Embed) IP Block Diagram**

The AXI4-Stream audio interface carries audio samples in AES3 format. The data width over the AXI4-Stream audio interface is fixed at 32-bit to carry one AES3 sub-frame as shown in the following figure. TID indicates the channel number of the audio sample data.

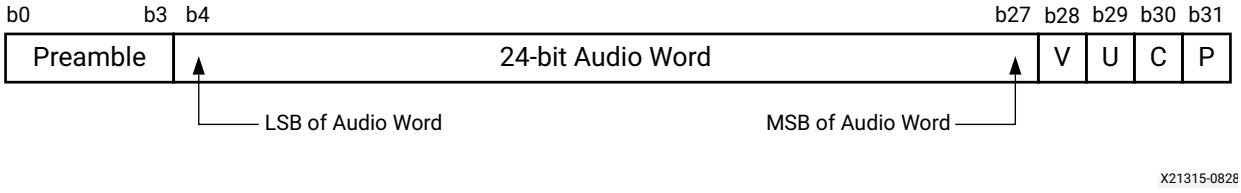

Figure 3-4: **AES3 Sub-frame Format with 24-Bit Audio Sample**

In HD-SDI mode, as per SMPTE ST 272, up to 16 channels of audio are inserted on data stream 1. In non-SD modes, as per SMPTE ST 299-1, up to 16 channels of audio are inserted onto data streams 1 and 2. In 3G, 6G and 12G SDI modes, as per SMPTE 299-1 and SMPTE ST 299-2, up to 32-channels of audio are inserted onto data streams 1, 2, 3 and 4. Audio control packets are inserted on data stream 1 and 3 (Y Video In) and data packets are inserted on data stream 2 and 4 (C_BC_R Video In). Remaining data streams from the SMPTE UHD-SDI TX Subsystem are delayed to match the latency of the audio insertion. Audio

embedded and latency matched data streams are sent back to the SMPTE UHD-SDI TX Subsystem for further processing.

The following figure highlights the SDI Interface to the audio embedder in 12G SDI mode with eight data streams. The audio embedder bypasses the non-audio ancillary packets before inserting the audio packets in the blanking region.

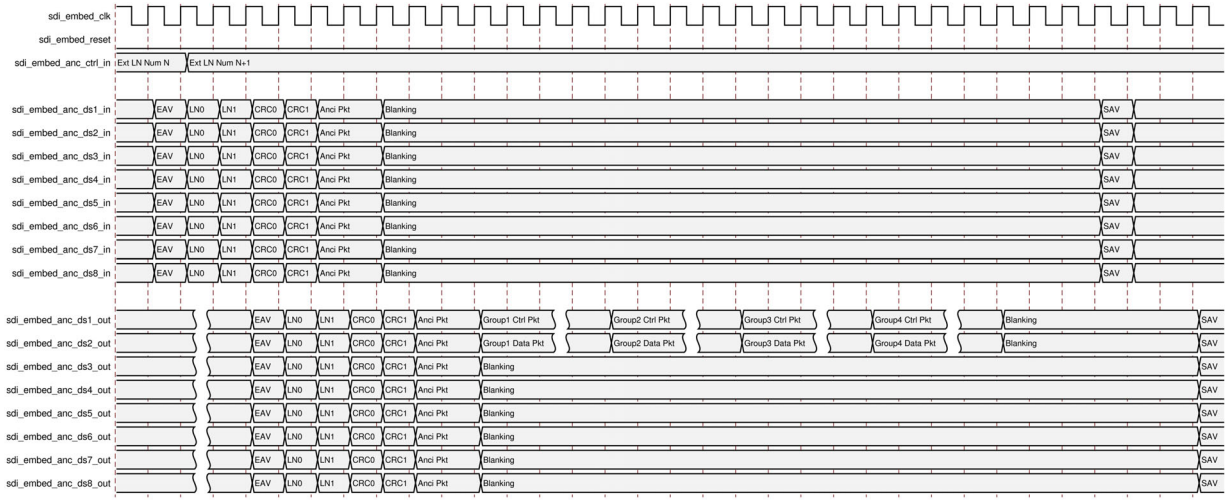

Figure 3-5: **SDI Interface Diagram of Audio Embedder**

IMPORTANT: *Audio samples should be placed on AXI4-Stream audio interface as per the audio sample rate. The Audio Embedder does not support burst audio samples that violate the audio sample rate from the external AXI4-Stream interface master.*

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 and 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 Subsystem

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

You can customize the IP for use in your design by specifying values for the various parameters associated with the IP subsystem 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 SMPTE UHD-SDI TX Subsystem.

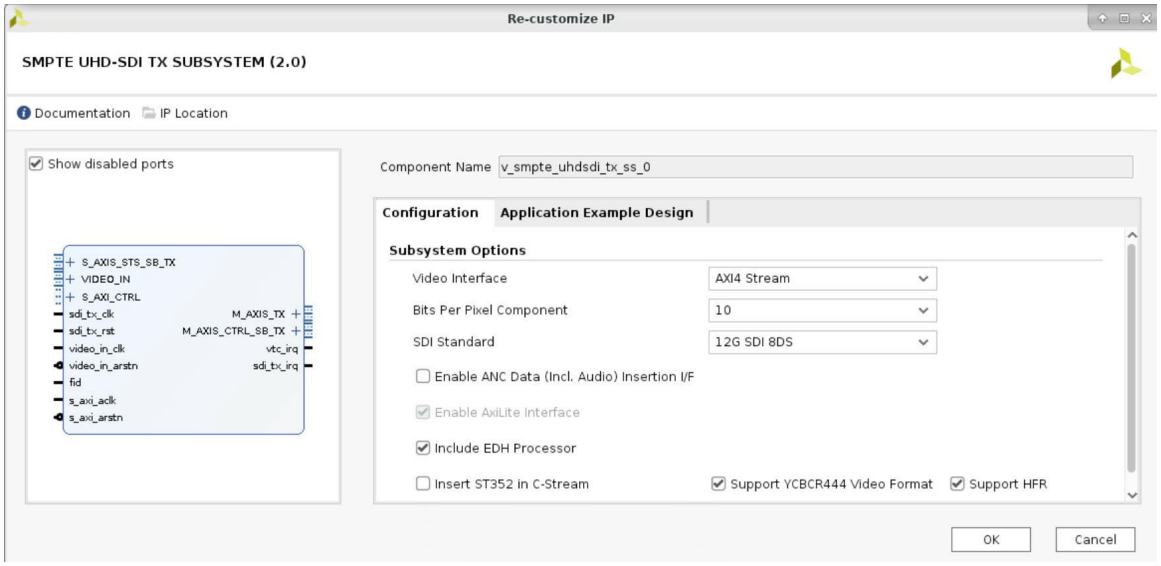

Figure 4-1: **Subsystem Configuration Tab**

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

- **Video Interface:** Specifies the user interface. The available options are:
  - AXI4-Stream
  - Native Video
  - Native SDI
- **Bit per Pixel Component**: Specifies the bit per component:
  - 10
  - 12
- **SDI Standard**: Specifies the SDI standard. The available options are:
  - 3G SDI
  - 6G SDI
  - 12G SDI 8DS
  - 12G SDI 16DS (available only for the Native SDI interface)
- **Enable ANC Data (Incl. Audio) Insertion I/F**: Allows the user to insert Ancillary data like Audio
- **Enable AxiLite Interface**: Includes AXI4Lite register interface. AXI Lite Interface is always enabled for AXI stream Interface. A user has the flexibility to choose the Enable AXI Lite option in case of native SDI or native Video Interface
- **Include EDH Processor**: EDH processing logic is included in the core
- **Insert ST352 into C-Stream**: Controls ST352 payload packets into C Stream
- **Support YCbCr444 Video Format**: The IP Subsystem supports YCbCr 444 Video Format only when this is enabled
- **Support HFR**: Includes HFR logic in the Core

Application Example Design Tab

Figure 4-2 shows the Application Example Design tab for the SMPTE UHD-SDI TX Subsystem pass-through example design. See Chapter 5, Example Design for more information on the example design.

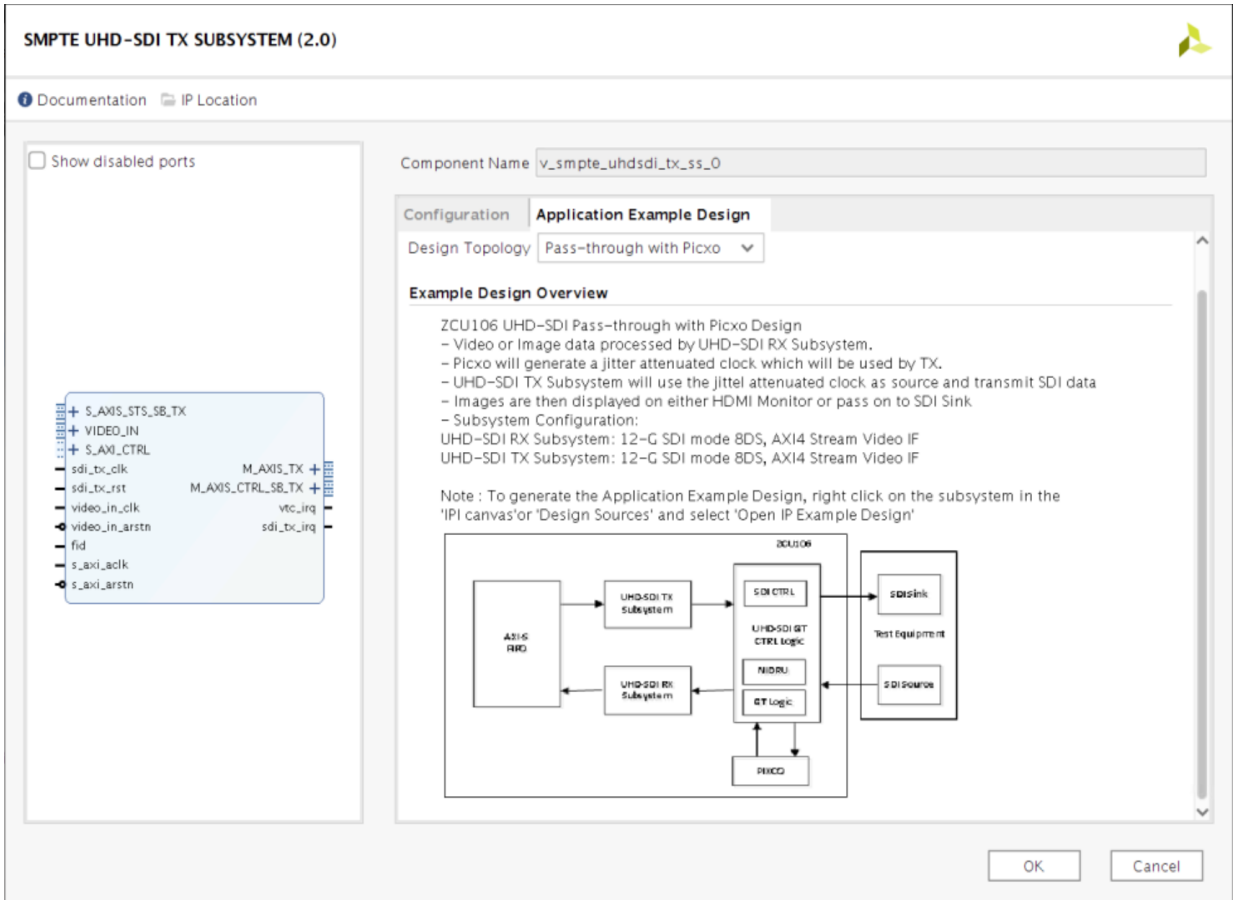

Figure 4-2: **Application Example Design Configuration Tab**

- **Target Board**: Target board on which the Application example design to be built. The available options are:
  - ZCU106 (Default)
- **Design Topology**: Type of configuration for Application Example Design. The available options are:
  - Pass-through with Picxo (Default)

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	**User Parameter/Value**	**Default Value**
Video Standard	C_VIDEO_INTF	AXI4-Stream
SDI Standard	C_LINE_RATE	12G SDI 8DS
Enable ANC Data (Incl. Audio) Insertion Interface	C_INCLUDE_ADV_FEATURES	FALSE
Enable AxiLite Interface	C_INCLUDE_AXILITE	TRUE
Include EDH Processor	C_INCLUDE_EDH	FALSE
ST352 into C stream	C_TX_INSERT_C_STR_ST352	FALSE
Bit per pixel component	C_BPP	10
Support HFR	C_INCLUDE_HFR	TRUE
Support YCbCr444 Video Format	C_INCLUDE_YCBCR_444	TRUE

Output Generation

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

Constraining the Subsystem

This section contains information about constraining the subsystem 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 TX 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].

Note: The SMPTE UHD-SDI TX Subsystem does not support simulation.

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].

Programming Sequence

The UHD-SDI TX Subsystem is delivered with a Baremetal driver and API designed to handle all programming of the subsystem core. This driver is automatically available in the Vitis™

tool when you create from a `.xsa` file, which includes the UHD-SDI TX Subsystem. This Baremetal driver is also available from the Xilinx Github [Ref 19]. The example designs in Chapter 5, Example Design showcase the use of the drivers in an application.

Information on Linux drivers can be found on the Xilinx Wiki [Ref 20].

Chapter 5

Example Design

This chapter contains information about the example design provided in the Vivado® Design Suite. Table 5-1 provides the hardware requirements for the example design.

Table 5-1: **Example Design Hardware Requirements**

Topology	Hardware	Processor	UHD-SDI GT Configurations		GT Type	GT Data Width	BPC
			TxPLL	RxPLL			
Pass-through with PICXO	ZCU106 2 HD-BNC to BNC cables SDI source and sink devices	A53	QPLL0	QPLL0	GTHE4	40-bit	10

Note: For additional pass-through designs, see [Ref 10].

ZCU106 UHD-SDI Pass-Through with PICXO Example Design

The UHD-SDI pass-through with the phase interpolator controlled crystal oscillator (PICXO) example design is built using the SMPTE UHD-SDI TX and RX Subsystems. Video or image data is received and processed by the UHD-SDI RX Subsystem. An AXI4-Stream FIFO is used for synchronization and temporary storage between the UHD-SDI RX Subsystem and the UHD-SDI TX Subsystems. The SMPTE UHD-SDI TX Subsystem transmits SDI data from the AXI4-Stream FIFO after the application programs the SMPTE UHD-SDI TX Subsystem sub-core registers based on the received SDI stream and the ST-352 payload packet data. The system is designed to replace external voltage-controlled crystal oscillator (VCXO) circuits by using functionality within each serial gigabit transceiver. For information on PICXO, see the *All Digital VCXO Replacement for Gigabit Transceiver Applications* (XAPP1241) [Ref 16].

Each transceiver has a phase interpolator (PI) circuit in the high-speed analog PLL output circuits that provides, on an individual transceiver channel basis, the ability to phase and frequency modulate the transmit clock operating the transceiver. Using a fully digital interface, the phase interpolator can be phase and frequency controlled from the device logic resources under control of a high-resolution programmable digital PLL. In conjunction with the device logic digital PLL, the phase interpolator provides the ability to phase or frequency modulate the transceiver data output directly locking to an input reference pulse or clock while providing a built-in clock cleaning filter function.

The example design application software runs on the Zynq® UltraScale+™ MPSoC Arm® processor subsystem (PS) and is fully software-controlled. (For simplicity, the PS is not shown in Figure 5-1.)

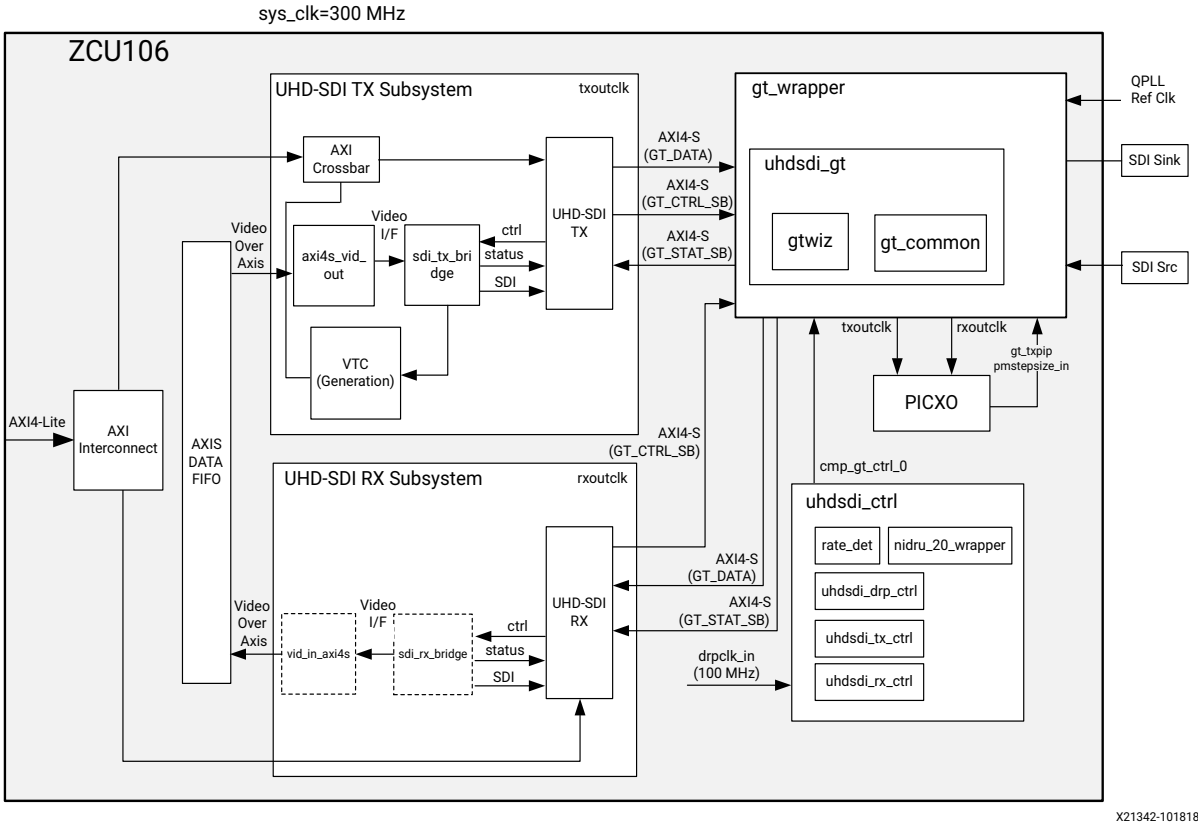

Figure 5-1: **ZCU106 UHD-SDI Pass-Through with PICXO Design**

Clocking

QPLL0 is allocated for both the UHD-SDI TX and RX datapath of the transceiver in the example design. Figure 5-2 shows the clocking used in this example design. The QPLL0 is given a single reference clock frequency, which can be either 148.5 MHz or 148.5/1.001 MHz, depending on which SDI line rate is to be supported (148.5 MHz for integer line rates or 148.5/1.001 MHz for fractional line rates) and comes from the on-board si570 chip.

In the example shown in Figure 5-2 below. The design only supports Integer Line rate, so the reference clock frequency is 148.5 MHz.

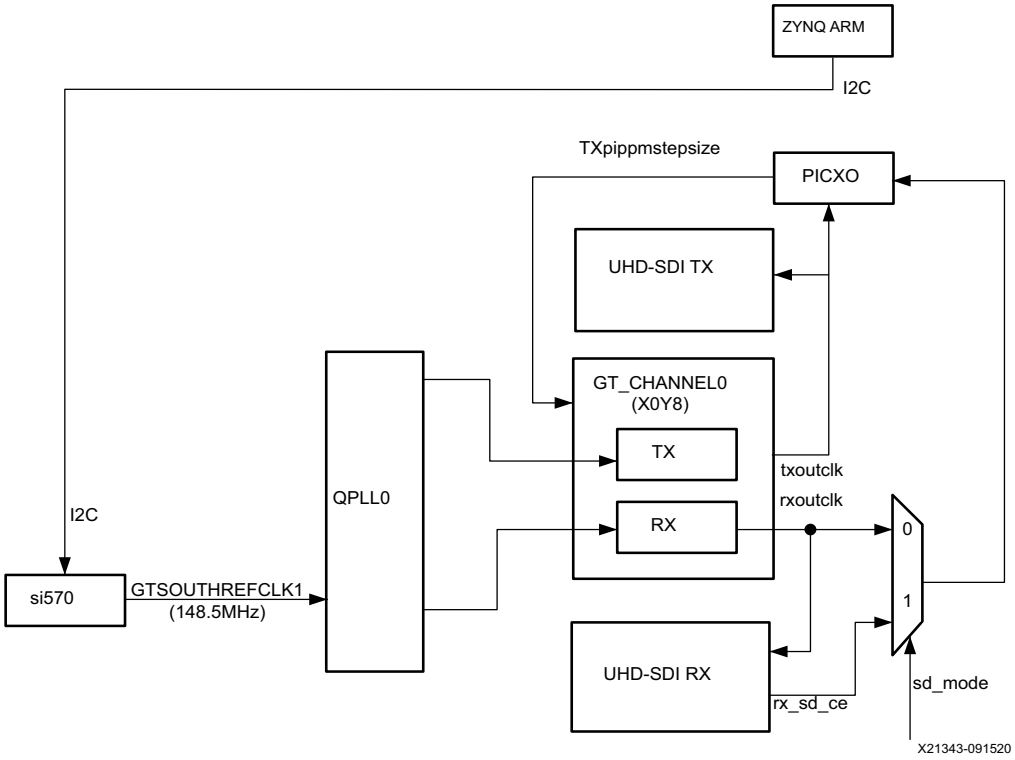

Figure 5-2: **ZCU106 UHD-SDI Pass-Through with PICXO Example Design Clocking**

Note: Design clocking is set up for a pass-through implementation and does not need the requirements from AR 72449.

Table 5-2 shows the clock frequency in different parts of the system for various SDI modes.

Table 5-2: **UHD-SDI example design clock Frequency Ranges**

SDI Mode	Tx_m/Rx_m	QPLL0 Ref clk (MHz)	txoutclk (MHz)(2)
SD-SDI	N/A	148.5	148.5
HD-SDI	0	148.5	74.25
HD-SDI	1	148.35	74.25/1.001
3G-SDI/6G-SDI	0	148.5	148.5
3G-SDI/6G-SDI	1	148.35	148.5/1.001
12G-SDI	0	148.5	297
12G-SDI	1	148.35	297/1.001

Notes:

1. For 12G-SDI, eight native SDI Data Streams (DS) is assumed.
2. For the transceiver TX, for integer and fractional frame rate, the PLL reference clock must be a different frequency: clock/1.000 for an integer frame rate and clock/1.001 for a fractional frame rate.

Example Design Transceiver Configuration

The example design uses the UHD-SDI GT (uhdsdi_gt_v2_0) core to configure the Xilinx® UltraScale+™ GTH transceivers and provides the option to select the transceiver reference clock. This core also generates control modules that are required to program the transceiver using the DRP interface. Figure 5-4 shows the Vivado® IDE configuration screen that is used in the ZCU106 application example design. See the *UHD-SDI GT LogiCORE IP Product Guide* (PG380) [Ref 15] for information on this core.

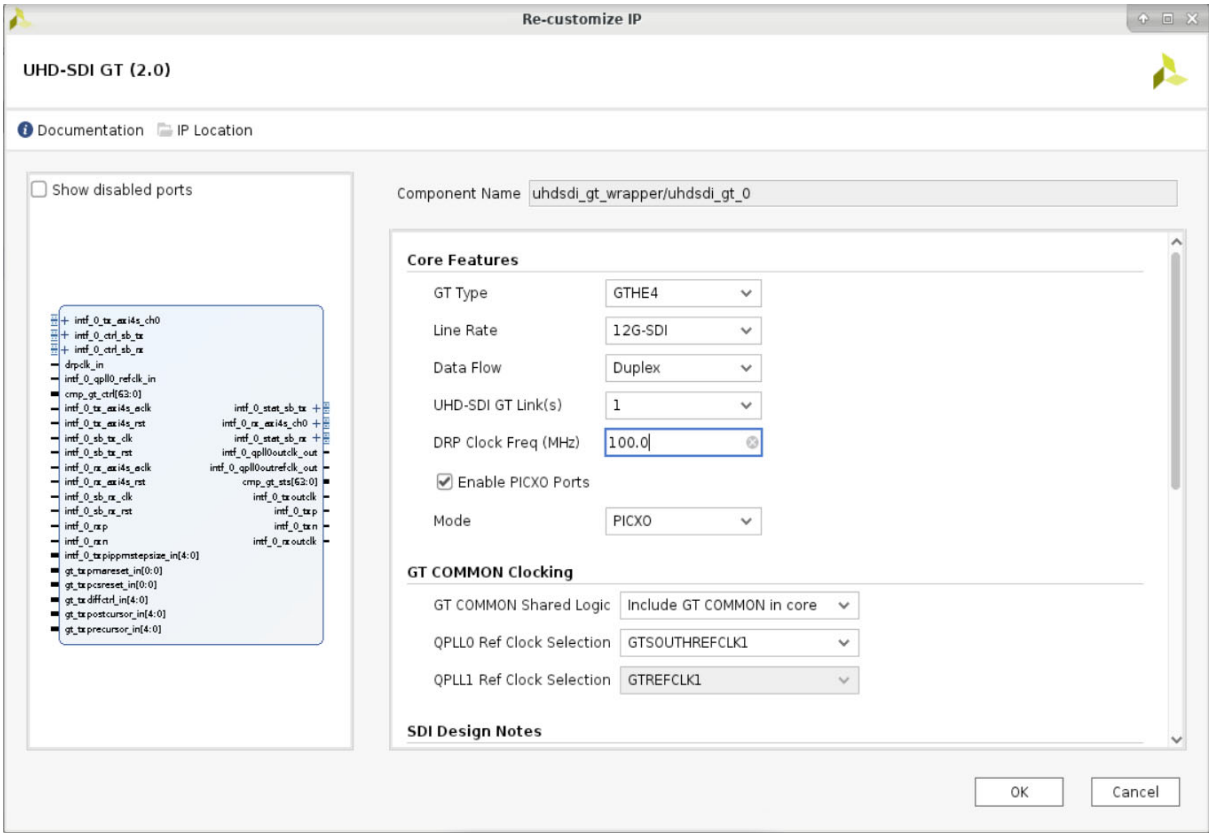

Figure 5-3: **(A) UHD-SDI GT Core Configuration**

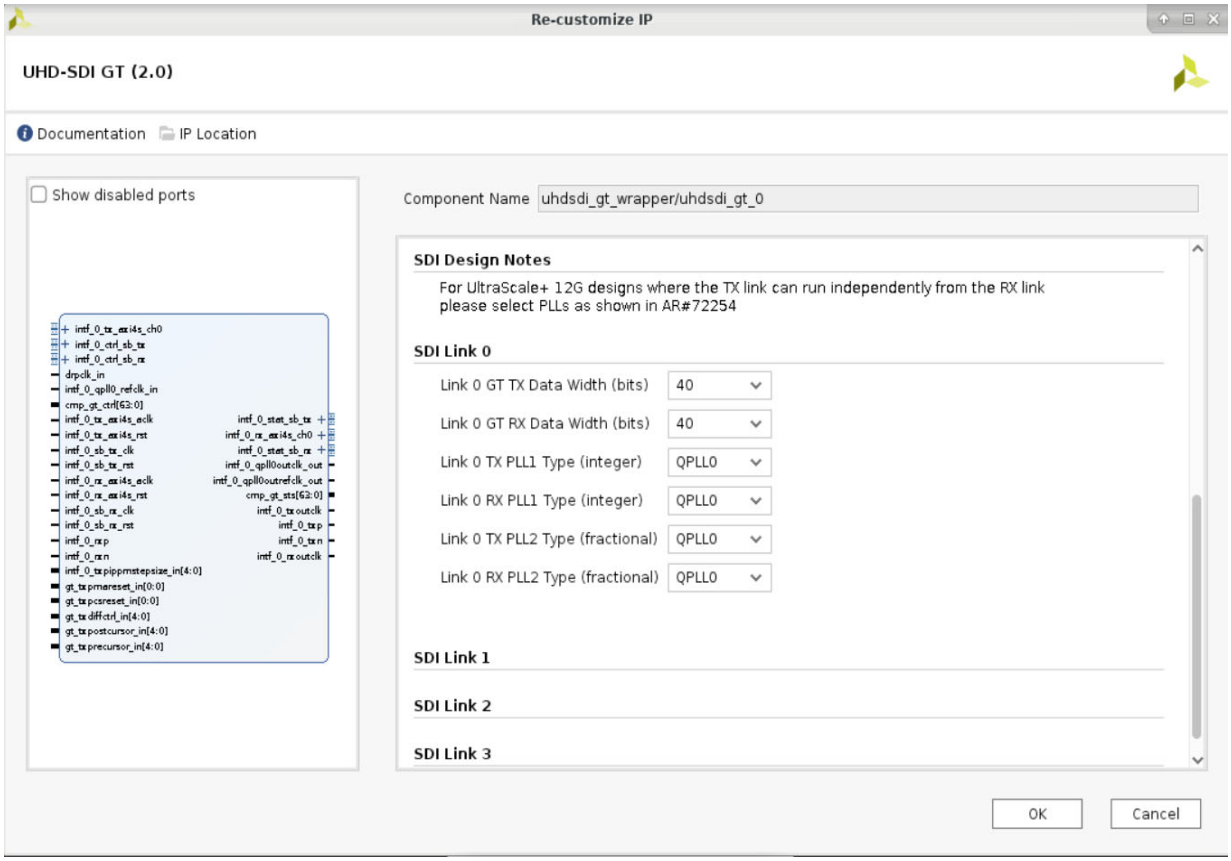

Figure 5-4: **(B) UHD-SDI GT Core Configuration**

Running the Example Design

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

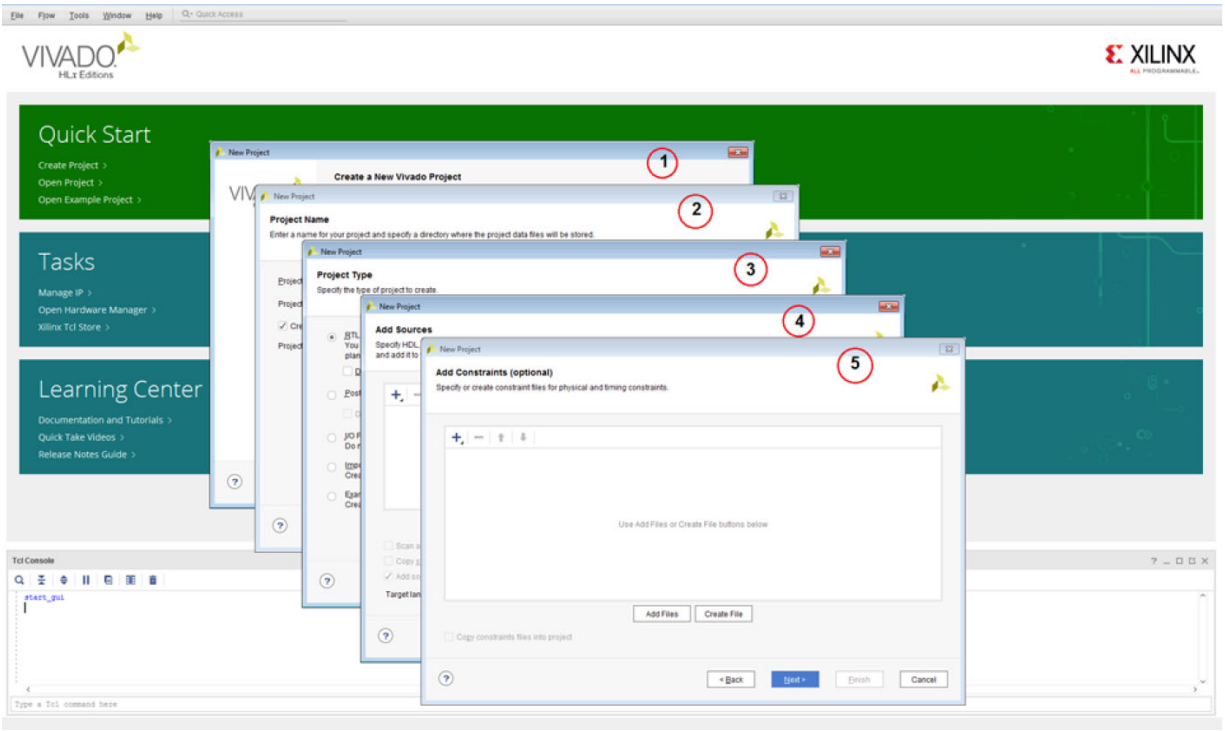

Figure 5-5: **Creating a New Project**

3. Select the board (ZCU106 supported), as shown below.

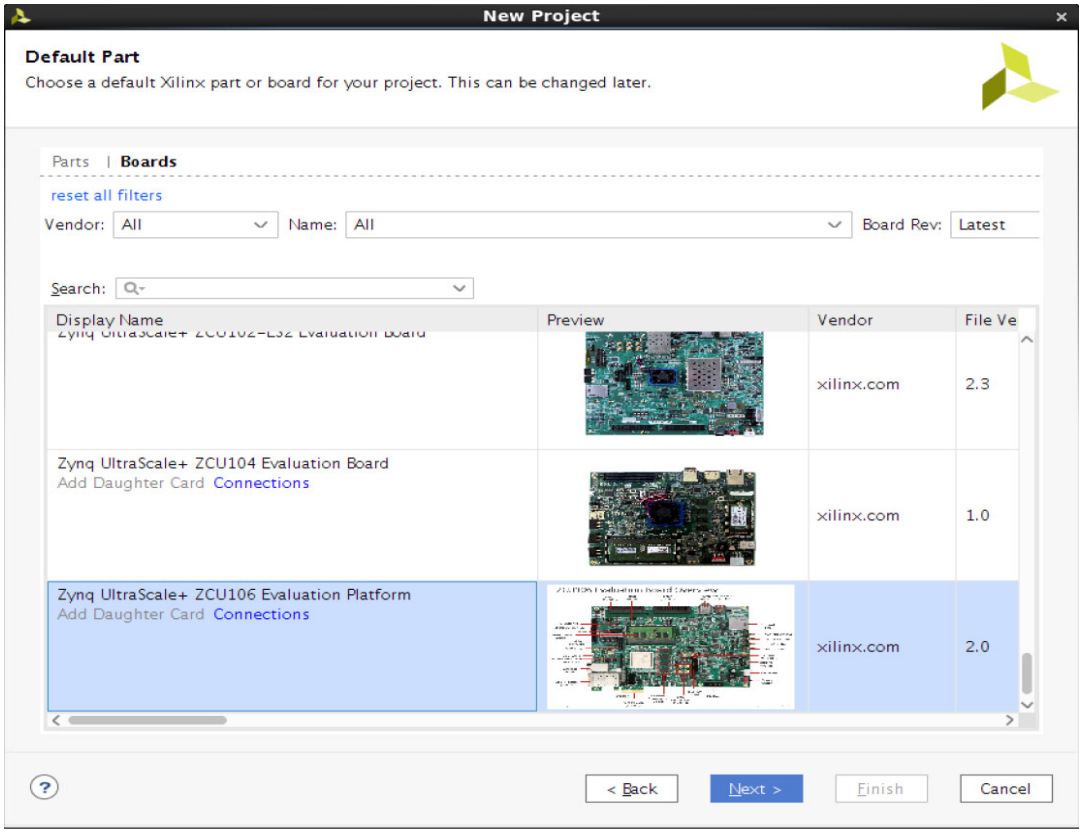

Figure 5-6: **Select the Board**

4. Click **Finish**.

5. Click **IP Catalog** and select the **SMPTE UHD-SDI TX Subsystem** under Video Connectivity, and double-click on it.

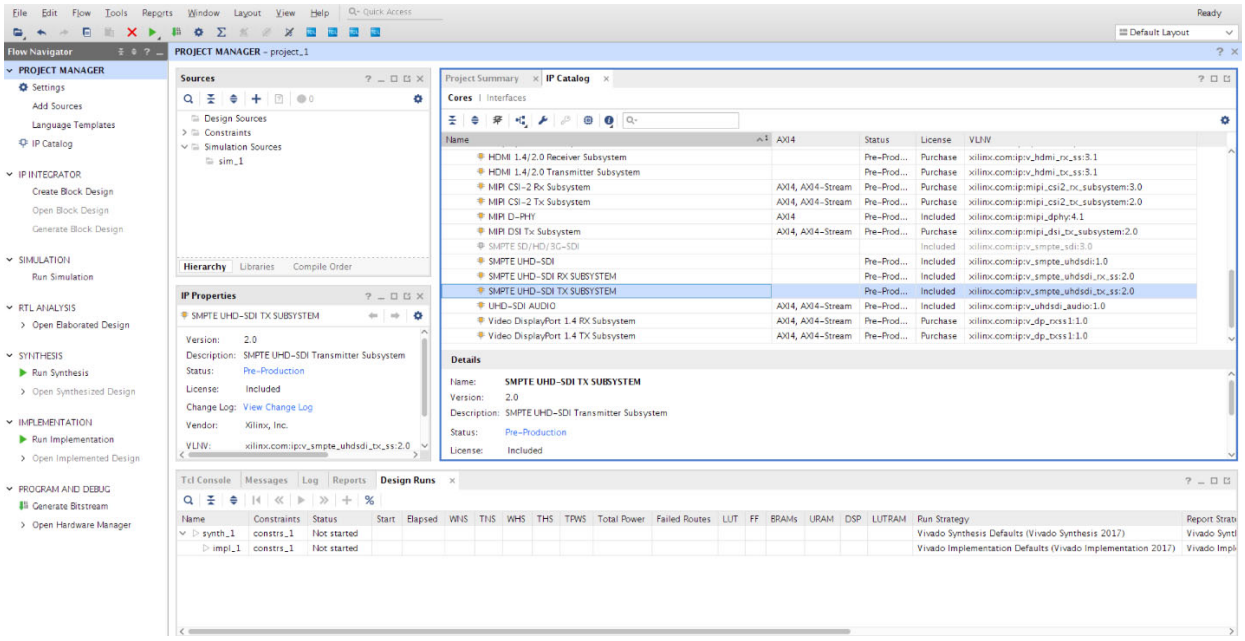

Figure 5-7: **Select the SMPTE UHD-SDI TX Subsystem**

For the Application Example Design flow, the IP configuration is based on the options selected in the Application Example Design tab. You can rename the IP component name, which is used as Application Example Design project name.

6. Configure the SMPTE UHD-SDI TX Subsystem Application Example Design tab, as shown in Figure 5-8.

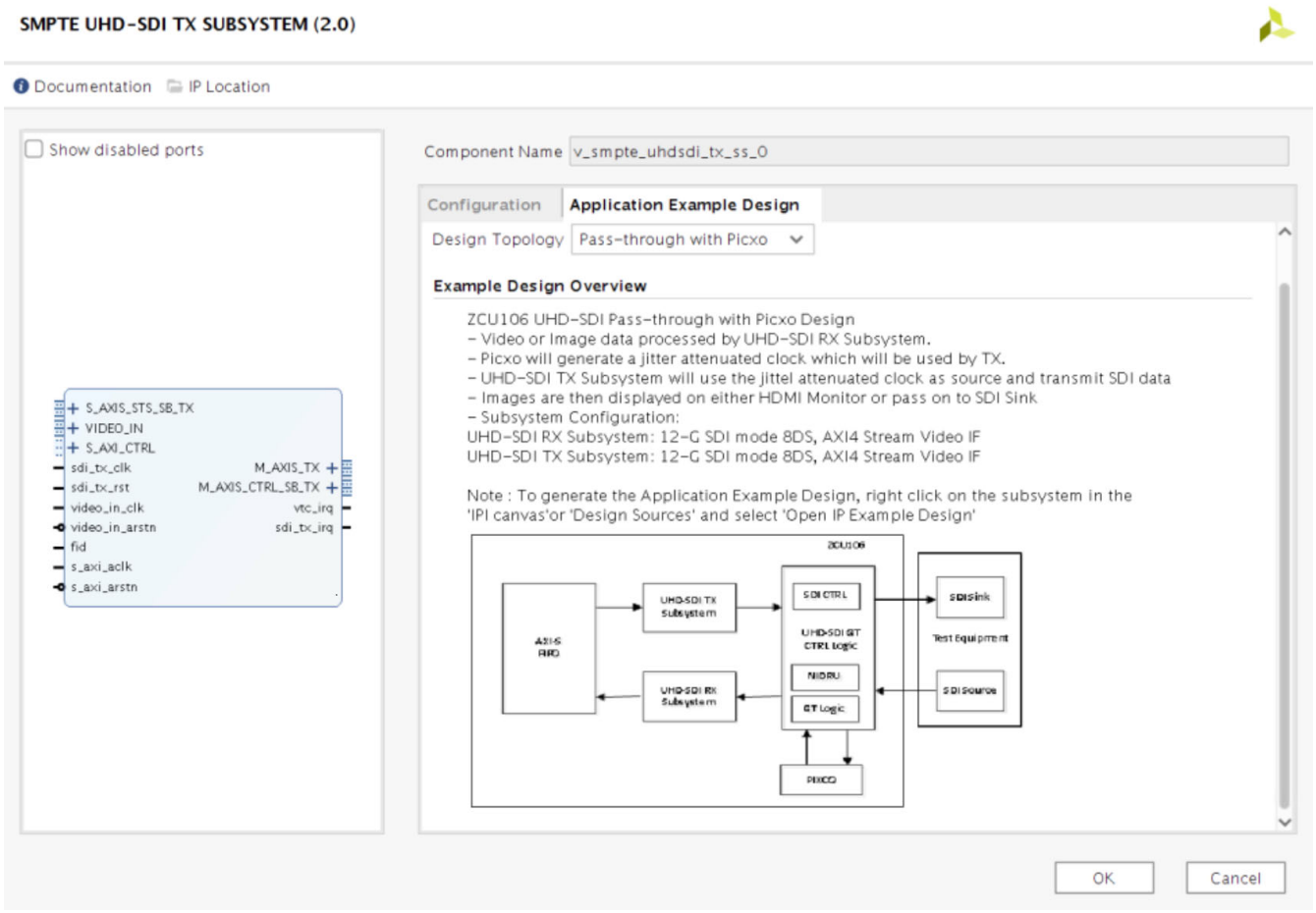

Figure 5-8: **SMPTE UHD-SDI TX Subsystem Application Example Design Tab**

7. Click **OK**. The Generate Output Products dialog box appears.

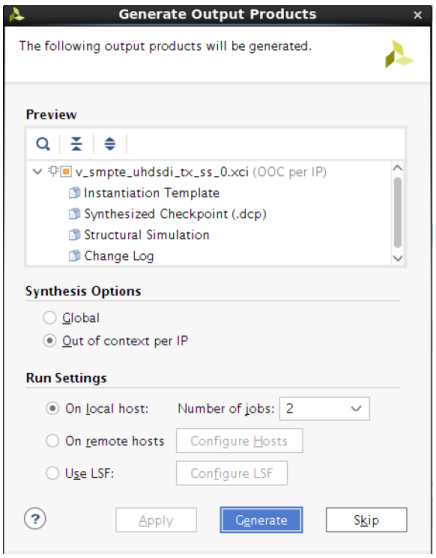

Figure 5-9: **Generate Output Products Dialog**

8. Click **Generate**.

   Note: You can optionally click **Skip** if you just want to generate the Application Example Design.

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

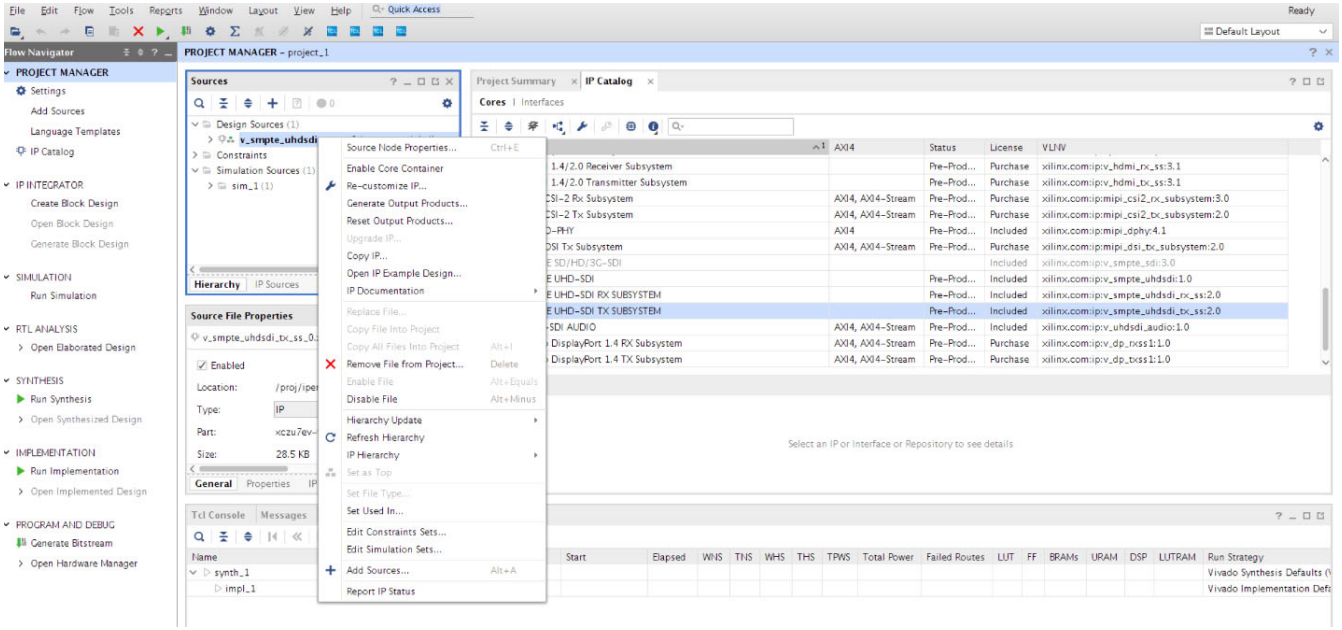

Figure 5-10: **Open the IP Sample Design**

10. Choose the target project location, and then click **OK**. The IP integrator design is then generated and creates the Vitis™ application. You can choose to Run Synthesis, Implementation, or Generate Bitstream. An overall system IP integrator block diagram of the ZCU106-based example design is as follows.

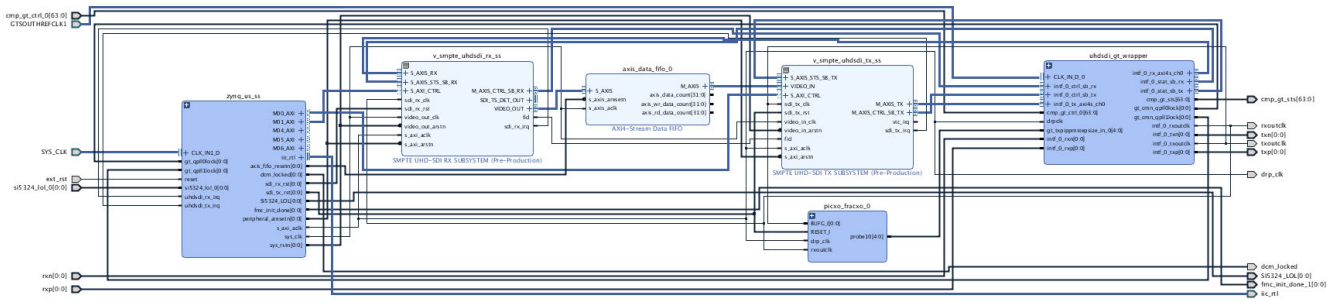

Figure 5-11: **Block diagram of the ZCU106-based Example Design**

Requirements

Hardware

The hardware requirements for this reference system are:

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

Software

This section includes any software requirements:

- Vivado Design Suite 2018.3 or later
- Vitis unified software platform 2019.2 or a later version
- Software terminals (for example, Tera Term, HyperTerminal, or PuTTY)

Setup

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

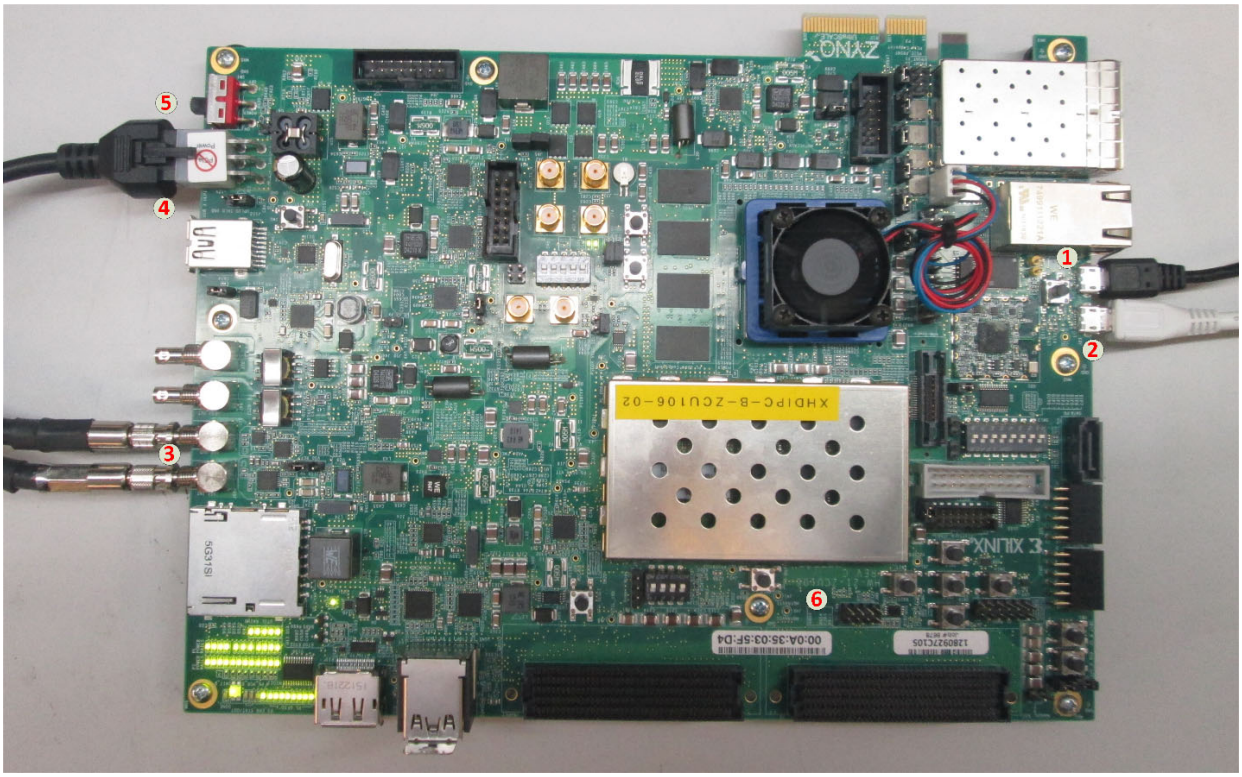

Figure 5-12: **ZCU106 Board Setup**

Note: In these instructions, the numbers in parentheses correspond to the callout numbers in Figure 5-12.

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_OUT link of ZCU106 (3) to the SDI sink device.
4. Connect the ZCU106 board to a power supply slot (4).
5. Switch on the ZCU106 board (5).
6. Make sure that the HW-ZCU106 board revision (6) is Rev C.
7. Start Tera Term or PuTTY to connect to the COM port interface 0 on the Host PC with 115200 bps, 8-bit, No parity, 1 stop bit, and no flow control as configuration.

Compiling Software in the Vitis Software Platform

The UHD-SDI Application example design generates the `.elf` file automatically. Use the following instructions:

1. In the Vivado Design Suite, click **Tools** -> **Launch Vitis**.
2. Select **Exported location** and **workspace** as `<Proj Dir>/<SW/xsdi_app>` and click **OK** to launch and open the Vitis project.

Running the Design on 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. Go to `<Component Name>_ex/imports`
3. Start the Xilinx Software Debugger (XSDB):

   ```
   source xsdb.tcl
   ```

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

The UART console displays the SDI stream details on the console.

UART Console Screens

Figure 5-13 shows the initial UART console output and menu options.

Figure 5-13: **Initial UART Console Output**

Information Option

Figure 5-14 shows the UART console output when the **i** key is pressed.

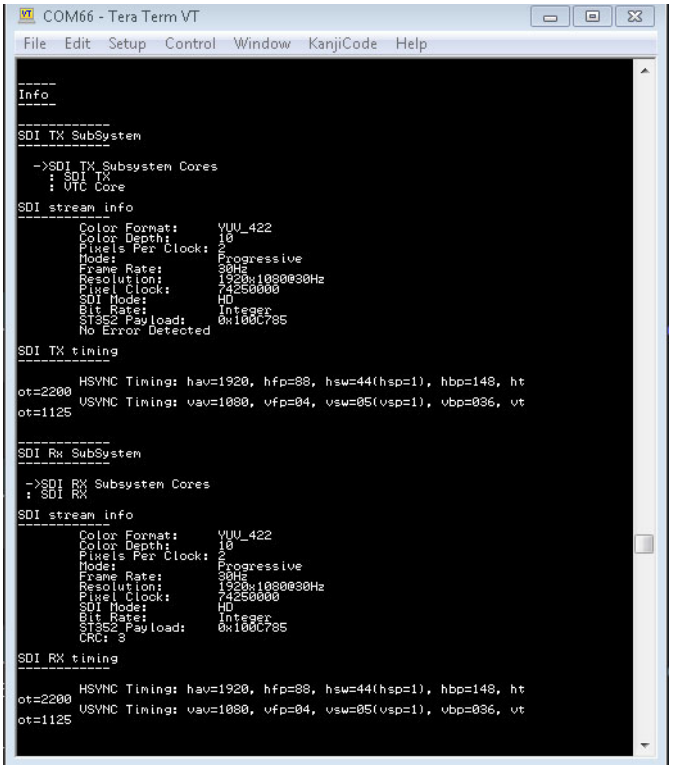

Figure 5-14: **UART Information Option**

Log Option

Figure 5-15 shows following figure shows the UART console output when the **z** key is pressed.

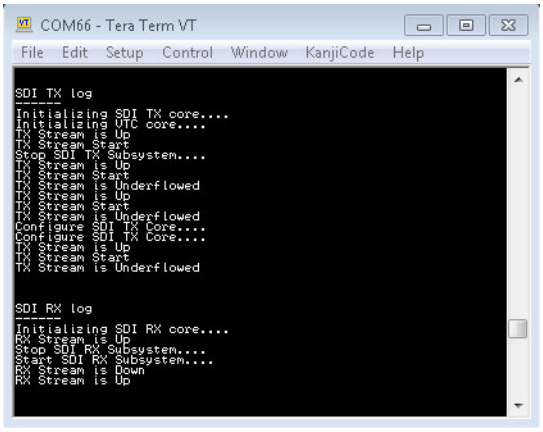

Figure 5-15: **UART Log Option**

Debug Option

Figure 5-13 shows following figure shows the UART console output when the **d** key is pressed.

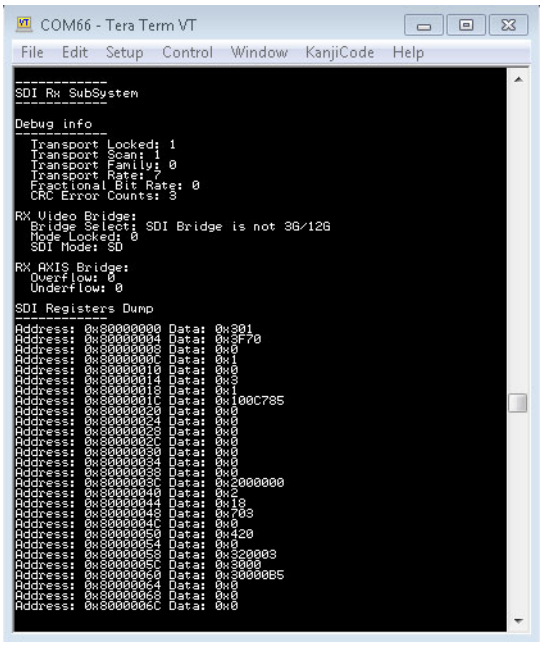

Figure 5-16: UART **Debug Option**

Versal ACAP Block Automation in UHD-SDI TX Subsystems

The Block Automation feature is provided in IP integrator to help you put together a basic system that connects the UHDSDI TX Subsystem IP with GT Quad using a GT Controller Bridge and a connection between Subsystem IP ports and external I/O ports.

To set up block automation:

1. Click on **Run Block Automation**, as shown in the following figure. Block automation connects the TX and RX data paths of the Parent IP to GT Wizard. (A new GT Wizard quad base is launched if it cannot pack the Parent IP with existing GT Quad resources.)

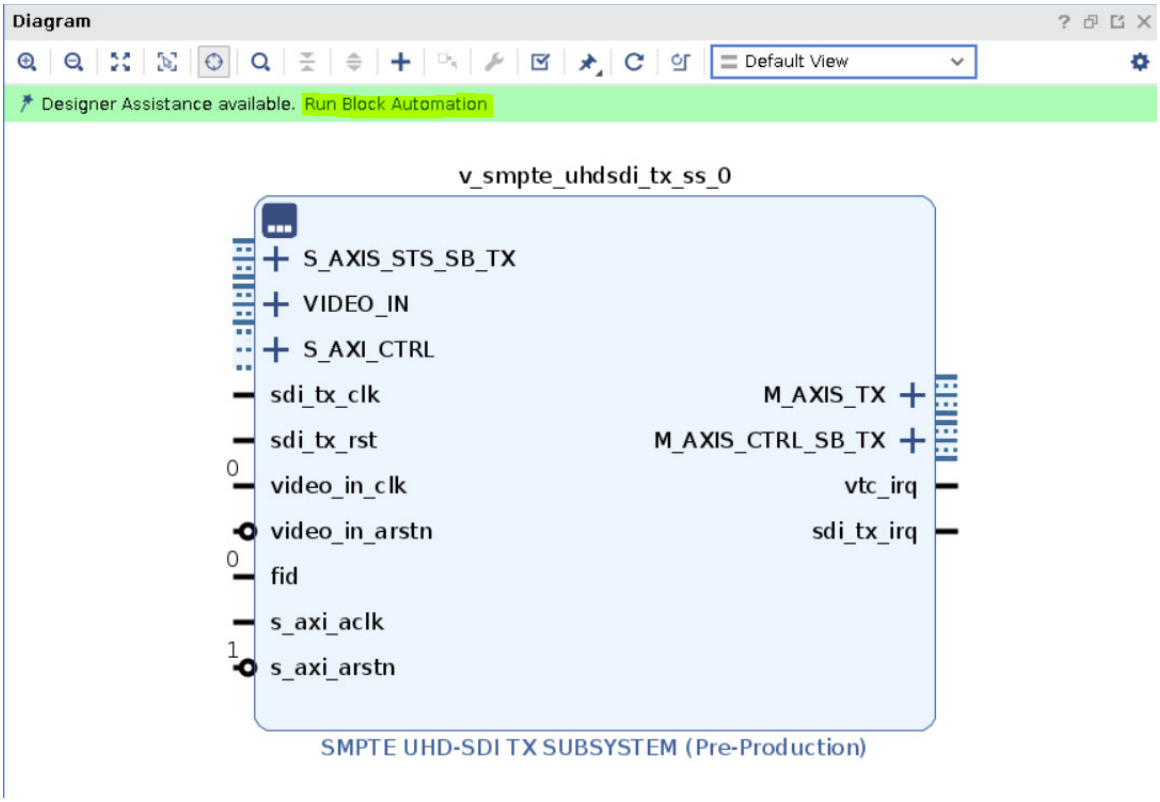

Figure 5-17: **Run Block Automation**

2. Use the Run Block Automation dialog box to specify the basic options that the UHDSDI TX subsystem IP needs. (Auto is selected by default.)

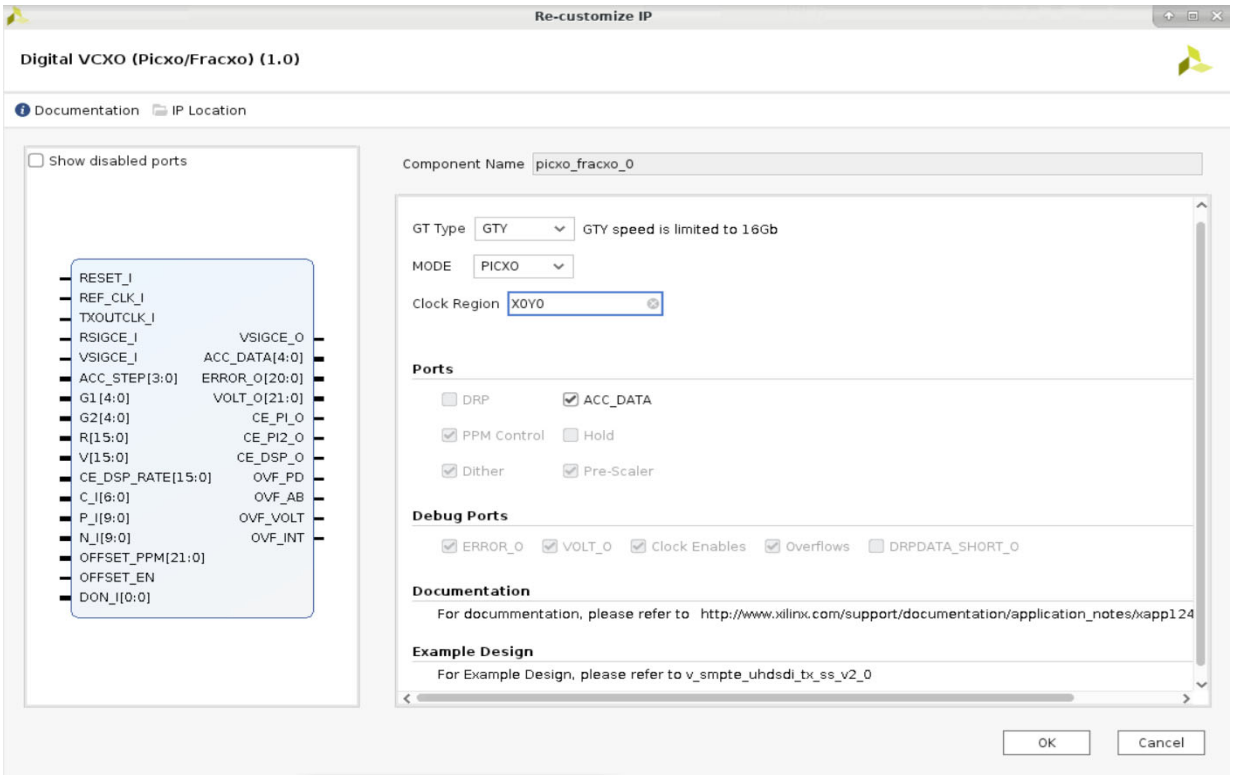

Figure 5-18: **Block Automation Options**

3. Click **OK**. The Block Automation feature then automatically creates a basic system as shown in the figure below. This example shows a basic system that consists of a UHDSDI TX Subsystem IP, a Versal® GT controller Bridge IP for SDI, the clock buffer, and the GT Quad for Versal® ACAP devices. Both clocks get connected to a clock source. (Because the design does not connect to any external I/O at this point, the IP integrator provides the Connection Automation feature as shown by the highlighted area in the figure.)

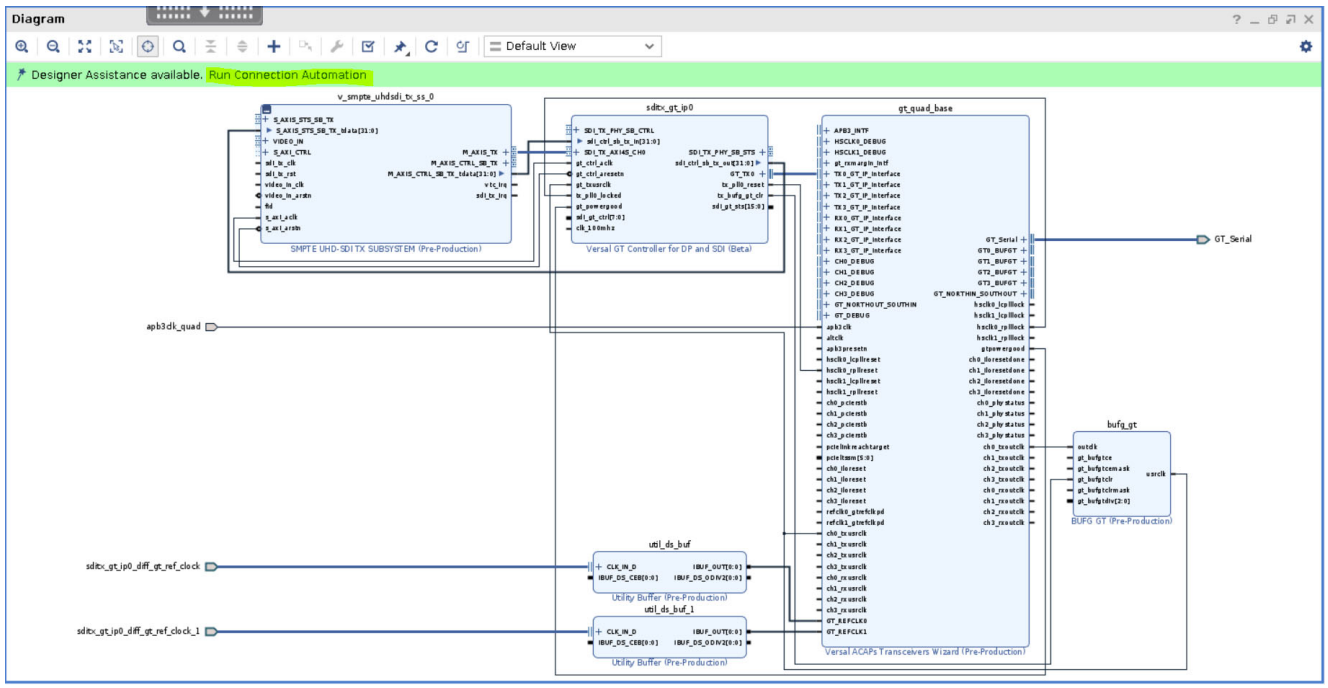

Figure 5-19: **Auto-connected Block, Basic System**

4. (Optional) Click on **Run Connection Automation** to get assistance in hooking interfaces and/or ports to external I/O ports. The following figure shows a list of the ports/interfaces that can use the Connection Automation feature.)

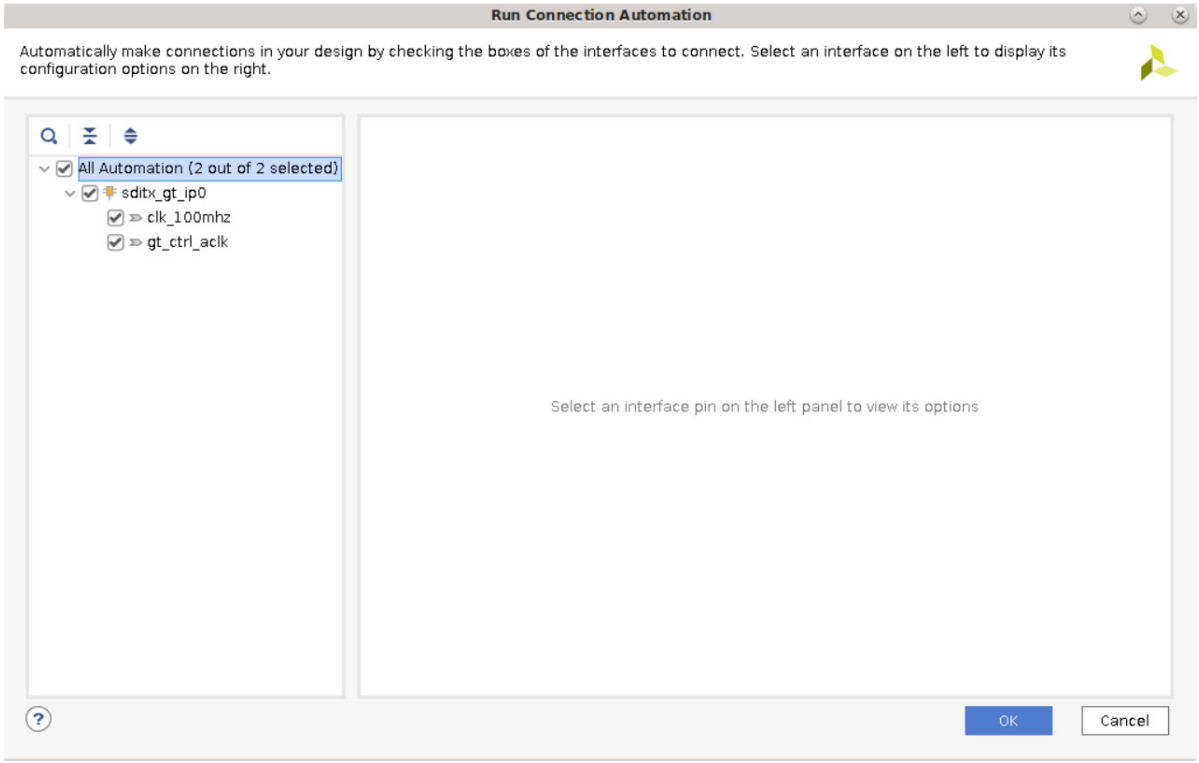

Figure 5-20: **Run Connection Automation**

5. Click **OK**. Both the clocks as shown in step 3 above are then connected to a clock source.

AMD XILINX

Appendix A

PICXO FRACXO IP Core

This appendix provides information about the PICXO FRACXO IP core that is used in UHD-SDI Subsystem example designs and provides details on clocking and use cases.

Customizing and Generating the Core

If you are customizing and generating the core in the Vivado® IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator* (UG994) [Ref 1] for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values do change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl Console.

The following table describes the IP specifics:

Table A-1: **IP Facts Table**

IP Specifics	
Supported Device Family(1)	UltraScale+™ (GTHE4, GTYE4) Versal® ACAP (GTYE5) Zynq® UltraScale+ MPSoC (GTHE4, GTYE4) Zynq UltraScale+ RFSoC
Supported User Interfaces	AXI4-Lite, AXI4-Stream, Native Video, Native SDI
Resources	Performance and Resource Utilization web page

Notes:
1. For a complete list of supported devices, see the Vivado IP catalog.

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 Vivado 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 Integrated Design Environment (IDE). The layout depicted here might vary from the current version.

You can customize the core using the following parameters, or allow defaults to be used.

Core Configuration Tab

Figure A-1 shows the Core Configuration tab for customizing the PICXO FRACXO core.

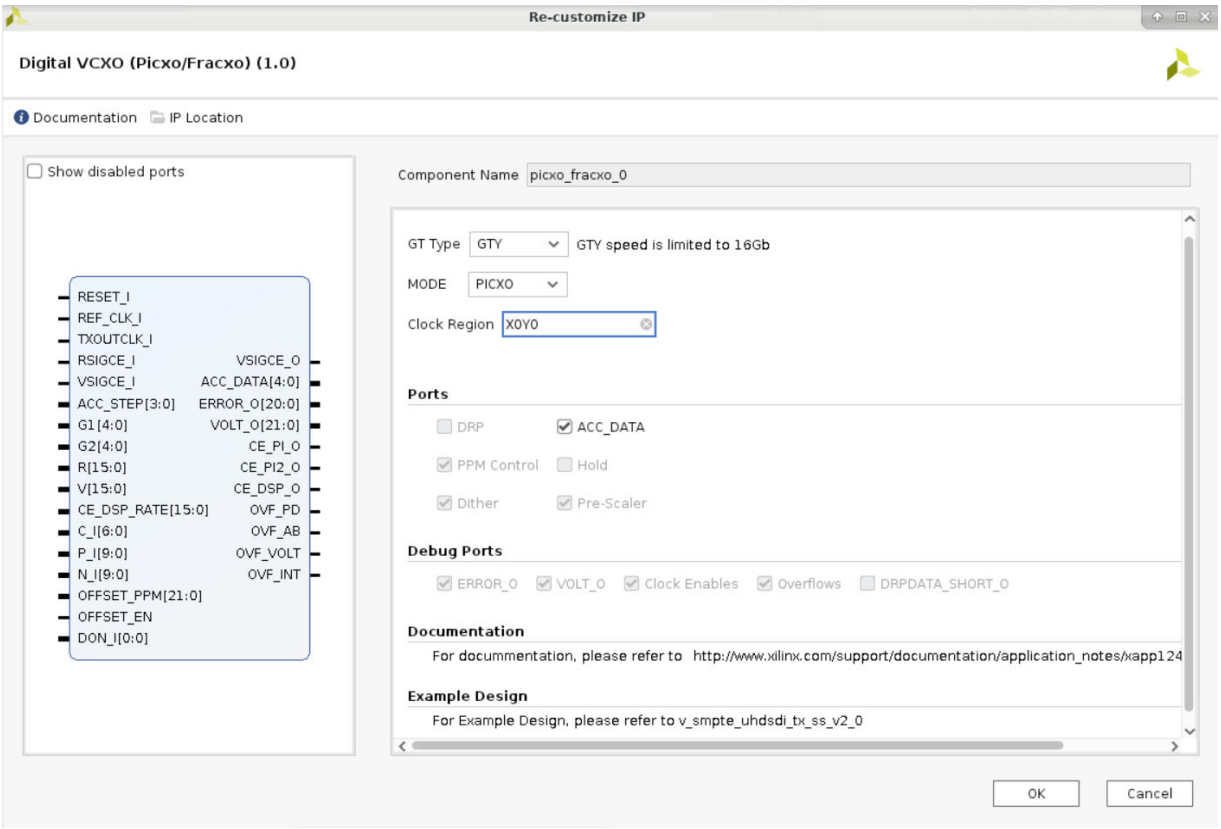

Figure A-1: **PICXO FRACXO Core Configuration Tab**

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

- **GT Type**: Select the GT type.
  - GTY
  - GTH
- **MODE**: Select the IP core mode
  - PICXO
  - FRACXO

Note: The IP provides the support for MODE as FRACXO but the feature is not validated on board.

- **Clock Region**: Specify the clock region for the GT.
  - X0Y0 (default)

Note: Provide the correct clock region and ensure that the transceiver(s) are present in the given clock region.

Ports

These settings allow selection of the core ports.

- **DRP**: Enables/disables the DRP ports in the core.
  - Disable (fixed)
- **ACC_DATA**: Enables/disables the ACC_DATA port which decides its step size. Range 1 to 15 (0 = no step).
  - Enable (default)
  - Disable
- **PPM Control**: Enables/disables OFFSET_PPM ports. It controls the direct frequency offset. OFFSET_PPM overwrites the output of the low-pass filter (VOLT_O) when OFFSET_EN is High.
  - Enable (fixed)
- **HOLD**: Assert to hold the Volt output at its current value.
  - Disable (fixed)
- **Dither**: Enables/disables DON_I port which can help to reduce jitter.
  - Enable (fixed)
- **Pre-Scaler**: Enables/disables pre-scaler.
  - Enable (fixed)

Debug Ports

These settings select the port to allow debugging of the core.

- **ERROR_O**: Enables/disables the ERROR_O port.

The ERROR output has an average value of around 0. This indicates that the DPLL has converged and is locked and the PICXO phase detector has nominally the same phase and frequency on its inputs.

  - Enable (fixed)

- **VOLT_O**: Enables/disables VOLT_O port.

The VOLT output has a value that represents the difference in frequency between the local crystal oscillator (XO) and the PICXO frequency-locked output. The greater the value from 0, the further in frequency the PICXO is tracking.

  - Enable (fixed)

- **Clock Enables**: Enables/disables the clock enable ports.
  - Enable (fixed)
- **Overflows**: Enables/disables the OVF_PD port which determines overflow in the phase detector.
  - Enable (fixed)

o **DRPDATA_SHORT_O**: Enables/disables the DRPDATA_SHORT port. DRPDATA_SHORT is not in use.

  - Disable (fixed)

User Parameters

Table A-2 shows the relationship between the fields in the Vivado IDE and the User Parameters (which can be viewed in the Tcl Console).

Table A-2: **Vivado IDE Parameter to User Parameter Relationship**

Vivado IDE Parameter	User Parameter	Default Value
MODE	MODE	PICXO
GT_TYPE	GT_TYPE	GTH
CLOCK_REGION	CLOCK_REGION	X0Y0
GT COMMON Shared Logic	SupportLevel	zynquplus
FAMILY	C_FAMILY	100.0
DRP	DRP	False
ACC_O	ACC_O	True

Table A-2: **Vivado IDE Parameter to User Parameter Relationship *(Cont'd)***

Vivado IDE Parameter	User Parameter	Default Value
OFFSET	OFFSET	True
HOLD	HOLD	False
DITHER	DITHER	True
PRESCALER	PRESCALER	True
OVF	OVF	True
ERROR	ERROR	True
VOLT	VOLT	True
CEs	CEs	True
DRPDATA_SHORT	DRPDATA_SHORT	True

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 core. 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 IP core.

Clock Frequencies

This section is not applicable for this IP core.

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking

This section is not applicable for this IP core.

Transceiver Placement

The PICXO FRACXO core places the transceivers in a clock region based on the clock region input.

I/O Standard and Placement

This section is not applicable for this IP core.

Clocking

Table A-3 provides details about the core clocks.

Table A-3: **Core Clocks**

Clock	Frequency	IP Configuration	Notes
drpclk_in	100.0 MHz (Default)	All	DRPCLK frequency value valid range differs for given device. See the respective data sheets for the clock range (FGTHDRPCLK for GTH transceiver).
rxoutclk	148.5 MHz for integer SDI line rate 148.35 MHz for fractional SDI line rate	All	RXOUTCLK clock from serial transceiver
BUFG_I (txoutclk)	148.5 MHz for integer SDI line rate 148.35 MHz for fractional SDI line rate	All	TXOUTCLK clock from serial transceiver. Connected to BUFG/BUFH/BUFR.

Reset

Table A-4 provides details about the core reset, and Interface to the Transceiver Ports.

Table A-4: **Core reset, and Interface to the Transceiver Ports**

Signal	Direction	Description
RESET_I	Input Synchronous reset	Active-High. Needs 8 clock cycles to reset correctly
DRPEN_O	Output	Unused. Leave floating
DRPWEN_O	Output	Unused. Leave floating

Table A-4: **Core reset, and Interface to the Transceiver Ports *(Cont'd)***

Signal	Direction	Description
DRPDO_I[15:0]	Input	Unused. Connect to 0 or 1
DRPDATA_O[15:0]	Output	Unused. Leave floating
DRPADDR_O[8:0]	Output	Unused. Leave floating
DRPRDY_I	Input	0 Unused. Connect to 0 or 1

Table A-5: **DRP User Port (Unused)**

Signal	Direction	Description
DRP_USER_REQ_I	Input	Unused. Connect to 0.
DRP_USER_DONE_I	Input	Unused. Connect to 0.
DRPEN_USER_I	Input	Unused. Connect to 0.
DRPWEN_USER_I	Input	Unused. Connect to 0.
DRPADDR_USER_I[8:0]	Input	Unused. Connect to 0.
DRPDATA_USER_I[15:0]	Input	Unused. Connect to 0.
DRPRDY_USER_O	Output	Unused. Leave floating.
DRPDATA_USER_O[15:0]	Output	Unused. Leave floating.
DRPBUSY_O	Output	Unused. Leave floating.

Table A-6: **TXPI Ports**

Signal Name	Direction	Description
ACC_DATA[4:0]	Output	Connect to TXPIPPMSTEPSIZE[4:0] of the transceiver.

Table A-7: **Debug Port**

Signal	Direction	Description
ERROR-O[20:0]	Output	Output of phase detector. Signed number.
VOLT_O[21:0]	Output	Output of low-pass filter. Signed number.
DRPDATA_SHORT_O[7:0]	Output	Unused. Leave floating.
CE_PI_O	Output	Clock enable for accumulator.
CE_PI2_O	Output	Clock enable for low pass filter and DAC.
CE_DSP_O	Output	Reset phase detector counters, load phase detector error into the low-pass filter.
OVF_PD	Output	Overflow in phase detector.
OVF_AB	Output	Saturation of the low-pass filter inputs.
OVF_INT	Output	Saturation of the low-pass filter integrator.
OVF_VOLT	Output	Saturation of the low-pass filter output.

Table A-8: **PICXO Loop Parameters**

Signal	Direction	Description
G1[4:0]	Input	Filter linear path gain: range `0` to `x12h`.
G2[4:0]	Input	Filter integrator path gain: range `0` to `x14h`.
R[15:0]	Input	Reference divider: range `0` to `65535`. Divides by R+2.
V[15:0]	Input	TXOUTCLK_I divider: range `0` to `65535`. Divides by V+2.
ACC-STEP[3:0]	Input	PICXO step size: range `1` to `15` (`0`=no step).
CE_DSP_RATE[15:0]	Input	DSP divider: default `07FF`. Control CE_DSP rate.
VSIGCE_I	Input	Clock enable to the TXOUTCLK_I divider. Connect to 1 for normal operation.
VSIGCE_O	Output	Reserved: Floating.
RSIGCE_I	Input	Clock enable of Reference divider. Connects to 1 for normal operation.
C_I[7:0]	Input	Reserved. Connect to `0`.
P_I[9:0]	Input	Reserved. Connect to `0`.
N_I[9:0]	Input	Reserved. Connect to `0`.
OFFSET_PPM[21:0]	Input	Direct frequency offset control. Signed number. OFFSET_PPM overwrites the output of the low-pass filter (VOLT_O) when OFFSET_EN is High.
OFFSET_EN	Input	Enable direct frequency offset control input. Active-High: Enables OFFSET_PPM input to overwrite output of low-pass filter (Volt).
HOLD	Input	Hold low-pass filter output value (Volt). Clock enable of Volt that stops Volt to the latest known ppm.
DON_I	Input	Dither On. Potential jitter reduction. Active-High.

AMD XILINX

Appendix B

Verification, Compliance, and Interoperability

The SMPTE UHD-SDI TX Subsystem has been verified on 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.

Hardware Testing

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

Hardware Validation

The SMPTE UHD-SDI TX Subsystem is tested in hardware for functionality, performance, and reliability using Xilinx® evaluation platforms. The SMPTE UHD-SDI TX 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 TX Subsystem has been validated using

- Zynq® UltraScale+™ MPSoC ZCU106 Evaluation Kit
- Kintex® UltraScale+ KCU116 Evaluation Kit
- Versal® ACAP VCK190 Evaluation Kit

The SMPTE UHD-SDI TX 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

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 TX, the Xilinx Support web page 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 TX. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the Downloads page. For more information about this tool and the features available, open the online help after installation.

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 TX Transmitter Subsystem

AR: 68767 (The master Answer Record contains patch updates, known issues, and guidance for the core.)

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 TX design issues. It is important to know which tools are useful for debugging various situations.

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature 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 devices.

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 5].

Hardware Debug

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.

- Does it work in post-place and route timing simulation? If problems are seen in hardware but not in timing simulation, this could indicate a PCB issue. Ensure that all clock sources are active and clean.
- Check MMCM lock and PLL lock signal(s) are asserted.
- Verify the I/O pin planning and XDC constraints.
- Follow recommended reset sequence.
- Verify all clocks are connected and that the frequencies are as expected.
- Enable the AXI4-Lite based register interface to get core status and control.
- Make sure that the serial line trace lengths are equal.
- Verify the FMC_VADJ voltage is 1.8V for FMC card use.

Transceiver (GT) Clocking

Note: For Versal® ACAP GT clocking information, refer to the *Versal ACAP GTY Transceivers Architecture Manual* [Ref 18].

- 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. See AR 57738 for debugging transceiver 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. See AR 57737 for more information.
- Measure `TXOUTCLK` and ensure that it is the expected frequency.
- Make sure the transceiver TXOUTCLK is the clock driving `tx_usrclk`, `TXUSRCLK`, and `TXUSRCLK2`.
- Monitor `TXBUFFSTATUS[2:0]` for overflow and underflow errors.

GT Initialization

- GTTXRESETDONE is asserted High after transceiver completes initialization.
- Ensure that the transceiver is not reset during normal operation.
- See AR 59435 for more information on debugging transceiver reset problems.
- Follow the recommended transceiver reset sequence.

Video Timing Controller (VTC) Debug

- Make sure that the VTC registers are programmed with the expected video timing parameters such as HACTIVE, VACTIVE, HTOTAL, VTOTAL, horizontal blanking, vertical blanking, etc,
- Make sure to program the VTC for Interlaced or progressive video mode.
- Check whether polarity of the signals (active-Low or active-High) are programmed correctly in VTC.

AXI4-Stream to Video Out Debug

- Check that the locked signal from the AXI4-Stream to Video Out core is asserted.
- Make sure that the overflow or underflow output signals are not asserted. If so, then check the connected clock frequencies and make sure that it matches the configured SDI mode line rate.
- Monitor `status[31:0]` to know the status of AXI4-Stream to Video Out core. This status bus is available in the AXI4S_VID_OUT_STS register of AXI4-Lite interface.

Video to SDI TX Bridge Debug

Make sure that the Video to SDI TX Bridge core is configured with the expected SDI mode value.

SMPTE UHD-SDI TX Core Debug

Make sure that theSMPTE UHD-SDI TX core is configured with the expected SDI mode value.

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 C-1 and Figure C-2. 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.

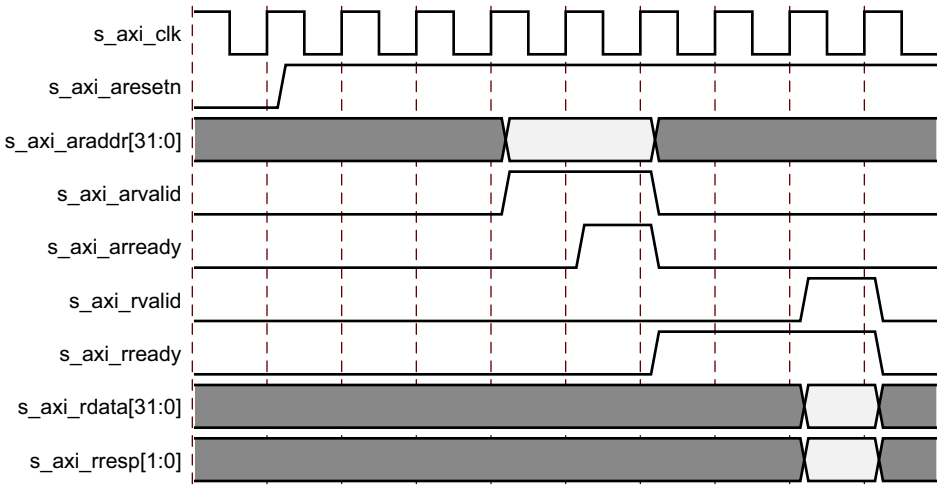

Figure C-1: **Read**

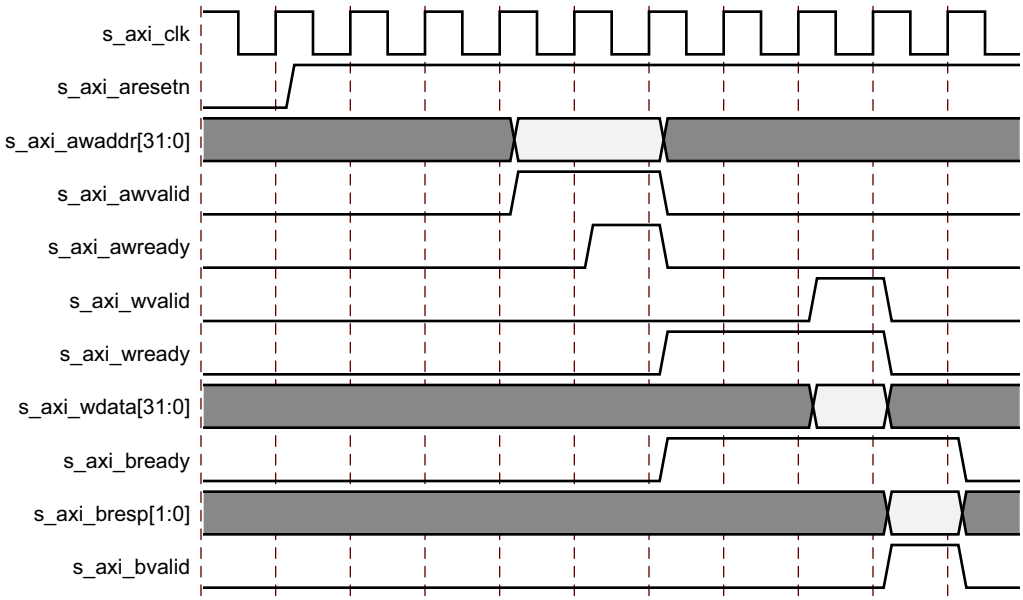

Figure C-2: **Write**

AMD XILINX

Appendix D

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. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
6. *SMPTE UHD-SDI LogiCORE IP Product Guide* (PG205)
7. *AXI4-Stream to Video Out Product Guide* (PG044)
8. *Video Timing Controller LogiCORE IP Product Guide* (PG016)
9. *AXI4-Stream Video IP and System Design Guide* (UG934)
10. *SMPTE UHD-SDI RX Subsystem Product Guide* (PG290)
11. *Vivado Design Suite: AXI Reference Guide* (UG1037)
12. *AXI SmartConnect LogiCORE IP Product Guide* (PG247)
13. *AXI Interconnect LogiCORE IP Product Guide* (PG059)
14. *UHD-SDI Audio Core IP Product Guide* (PG309)
15. *UHD-SDI GT LogiCORE IP Product Guide* (PG380)
16. *All Digital VCXO Replacement for Gigabit Transceiver Applications* (XAPP1241)
17. *Versal AI Core Series Data Sheet: DC and AC Switching Characteristics* (DS957)
18. *Versal ACAP GTY Transceivers Architecture Manual* (AM002)
19. Xilinx Github (https://github.com/Xilinx)
20. Xilinx Wiki (https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18841996/Linux)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/19/2022	2.0	Updated Unsupported Features, Native SDI Signals, and Native Video (VID_IO_IN) Interface Signals sections.
05/18/2022	2.0	Editorial changes.
04/26/2022	2.0	Added parameters to select HFR and YCbCr444 support to reduce resource count.
06/30/2021	2.0	• Added HFR Support for 10-bit 6G/12G mode at AXI Interface Subsystem Level • Updated Unsupported Features section • Updated module control register description for HFR • Added PICXO FRACXO IP Core for PICXO IP in Appendix A
01/11/2021	2.0	• Added Versal ACAP support. • Added Versal ACAP Block Automation in UHD-SDI TX Subsystems section. • Added new Programming Sequence section. • Updated Unsupported Features section. • Updated Compiling Software in the Vitis Software Platform section.
11/18/2020	2.0	Added Support for HLG HDR in IP Facts section.
09/15/2020	2.0	• Updated 12-bit and HFR support in the following sections: ◦ IP Facts ◦ Unsupported Features ◦ Core Configuration Tab ◦ User Parameters • Updated Clocking section in the Example Design chapter. • Updated Clocking section in the Designing with the Core chapter. • Added information on CPLL-QPLL usage for 12G SDI in the CPLL Clocking section.
06/14/2019	2.0	Updated Figure 5-4.
12/05/2018	2.0	• SMPTE 352: Payload packet insertion of Y stream and C stream supported ◦ Native SDI Signals ◦ SMPTE UHD-SDI TX Registers ◦ MODULE_CTRL Register (0x04) ◦ SS_CONFIG Register (0x40) • Application Example Design Tab added • ZCU106 UHD-SDI Pass-Through with PICXO Example Design Example Design added • Appendix A, PICXO FRACXO IP Core added
04/04/2018	2.0	• Added Native Video and Native SDI interfaces • Added YCbCr420 format feature details
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 https://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 https://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 2017–2022, Advanced Micro Devices, Inc., Xilinx, the Xilinx logo, Alveo, Artix, Kintex, Spartan, Versal, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. AMBA, AMBA Designer, Arm, ARM1176JZ-S, CoreSight, Cortex, PrimeCell, Mali, and MPCore are trademarks of Arm Limited in the EU and other countries. All other trademarks are the property of their respective owners.