SMPTE UHD-SDI Transmitter Subsystem v1.0

LogiCORE IP Product Guide

Vivado Design Suite

PG289 October 4, 2017

XILINX
ALL PROGRAMMABLE™

Table of Contents

Introduction

The Society of Motion Picture and Television Engineers (SMPTE) UHD-SDI transmitter subsystem implements a SDI transmit interface in accordance to the serial digital interface (SDI) family of standards. The subsystem accepts video from AXI-4 Stream Video interface and outputs Native Video stream, and allows for fast selection of the top-level parameters and automates most of the lower level parameterization. The AXI4-Stream video interface allows a seamless interface to other AXI4-Stream-based subsystems.

Features

- Support for 2 pixel per sample
- 10-bit per color component
- Supports YUV 4:2:2 color space
- AXI4-Lite interface for register access to configure different subsystem options
- Standards compliance:
  - SMPTE ST 259: SD-SDI at 270 Mb/s
  - SMPTE 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.
  - SMPTE ST 352: Payload ID packets are fully supported

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

Notes:

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

Overview

Introduction

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

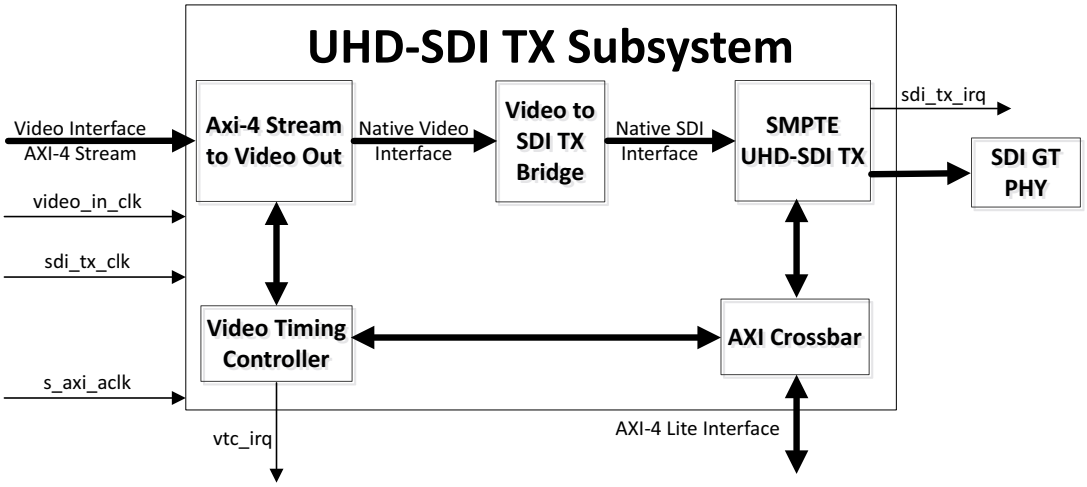

Figure 1-1: **Transmitter Subsystem Architecture**

The subsystem consists of the following subcores:

- AXI-4 Stream to Video Out
- Video to SDI TX Bridge
- SMPTE UHD-SDI TX
- Video Timing Controller
- AXI Crossbar

Subcore Details

AXI-4 Stream to Video Out

The AXI-4 Stream to Video Out core act as an interface from the AXI4-Stream interface implementing a Video Protocol to a video source (parallel video data, video syncs, and blanks).This core works with the Xilinx Video Timing Controller (VTC) core. See *AXI-4 Stream to Video Out Product Guide* [Ref 9] for details.

Video to SDI TX Bridge

The LogiCORE IP Video to SDI TX Bridge connects the video output of the AXI4-Stream to Video Output core to the SDI transmitter input of the SMPTE SDI TX core. 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 1-2 shows the top level bridge architecture.

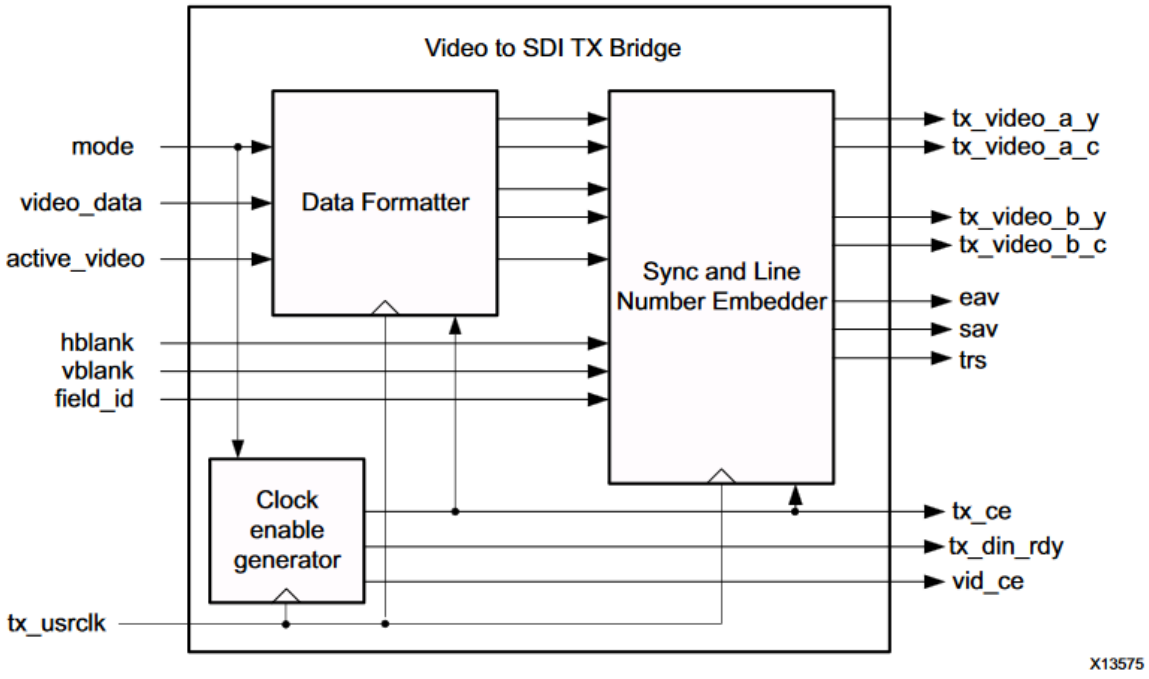

Figure 1-2: **Top-Level Block Diagram of Video to SDI TX Bridge**

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 bits per component. For SDI-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 SDI TX bridge and generates single multiplexed SDI 10-bit data stream. See the *SMPTE UHD-SDI Product Guide* [Ref 8] for details.

Video Timing Controller

The Video Timing controller core is used to generate the Video timing and used by AXI-4 Stream to Video out core for native video interface signals generation. See the *Video Timing Controller LogiCORE IP Product Guide* [Ref 10] 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* [Ref 14] for details.

Applications

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

Unsupported Features

- 16-way data stream interleaving is not supported

Licensing

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

Product Specification

Standards

The core supports the following SMPTE standards:

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

  Dual link and quad link 6G-SDI and 12G-SDI are supported by instantiating two or four UHD-SDI cores.
- SMPTE ST 352: Payload ID packets are fully supported.

Performance

Maximum Frequencies

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

Resource Utilization

For full details about performance and resource utilization, visit the Performance and Resource Utilization web page.

Port Descriptions

The SMPTE UHD-SDI TX Subsystem I/O signals are described in Table 2-1.

Table 2-1: **Port Descriptions**

Signal	Direction	Description
AXI-4 Lite Interface Signals (Enable AxiLite Interface Option Selected)		
s_axi_aclk	Input	AXI-4 Lite clock
s_axi_arstn	Input	AXI-4 Lite reset. Active-Low
S_AXI_CTRL*		AXI4-Lite interface, defined in the *Vivado Design Suite: AXI Reference Guide* (UG1037) [Ref 13]
Video-Over-AXIS Interface Signals (Enable Vid-Over-AXI4S Interface Option Selected)		
video_in_clk	Input	Video input clock
video_in_arstn	Input	Video input active-Low reset.
VIDEO_IN_tdata[63:0]	Output	Video input data for carrying YUV 4:2:2 video with 10 bpc. (for details refer to PG044, AXI4-Stream Data Interface Signal Descriptions)
VIDEO_IN_tlast	Output	AXI4-Stream TLAST. End of Line
VIDEO_IN_tready	Input	AXI4-Stream TREADY.
VIDEO_IN_tuser	Output	AXI4-Stream TUSER. Start of Frame
VIDEO_IN_tvalid	Output	AXI4-Stream TVALID. Active video data enable
fid	Output	Field ID
S_AXIS_STS_SB_TX Interface Signals		
S_AXIS_STS_SB_TX_tready	Output	Core Ready
S_AXIS_STS_SB_TX_tvalid	Input	Data valid
S_AXIS_STS_SB_TX_tdata[31:0]	Input	Sideband signal information from transceiver block
S_AXIS_TX Interface Signals		
sdi_TX_clk	Input	SMPTE SDI TX Core clock
sdi_TX_rstn	Input	Active low reset
S_AXIS_TX_tready	Output	SMPTE SDI TX Core ready
S_AXIS_TX_tvalid	Input	Data valid

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

Signal	Direction	Description
S_AXIS_TX_tdata[n-1:0]	Input	n is varies with SDI standard selection. n=40 for 6G-SDI & 12G-SDI n=20 for 3G-SDI
S_AXIS_TX_tuser[31:0]	Input	TUSER Information
M_AXIS_CTRL_SB_TX Interface Signals		
M_AXIS_CTRL_SB_TX__tready	Output	Core Ready
M_AXIS_CTRL_SB_TX_tvalid	Input	Data valid
M_AXIS_CTRL_SB_TX_tdata	Input	Sideband signal information from transceiver block
Interrupt Signal		
sdi_tx_irq	Output	SMPTE UHD-SDI TX core interrupt
vtc_irq	Output	VTC core interrupt
SMPTE UHD-SDI TX Core Signals(1) (Enable Vid-Over-AXI4S Interface Option NOT Selected)		
sdi_tx_ctrl[31:0]	Input	Bit0: module_enable; Bit1: not used; Bit3~bit2: reserved; Bit6~bit4: tx_mode: 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; Bit31~bit21: reserved;
ST352_DATA_IN_tx_st352_data_ch0[31:0]	Input	ST352 data for channel 0
ST352_DATA_IN_tx_st352_data_ch1[31:0]	Input	ST352 data for channel 1

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

Signal	Direction	Description
ST352_DATA_IN_tx_st352_data_ch2[31:0]	Input	ST352 data for channel 2
ST352_DATA_IN_tx_st352_data_ch3[31:0]	Input	ST352 data for channel 3
ST352_DATA_IN_tx_st352_line_f1	Input	Odd line to insert ST352 data
ST352_DATA_IN_tx_st352_line_f2	Input	Even line to insert ST352 data
SDI_DS_IN_ds1[9:0]	Input	SDI data stream 1
SDI_DS_IN_ds2[9:0]	Input	SDI data stream 2
SDI_DS_IN_ds3[9:0]	Input	SDI data stream 3
SDI_DS_IN_ds4[9:0]	Input	SDI data stream 4
SDI_DS_IN_ds5[9:0]	Input	SDI data stream 5
SDI_DS_IN_ds6[9:0]	Input	SDI data stream 6
SDI_DS_IN_ds7[9:0]	Input	SDI data stream 7
SDI_DS_IN_ds8[9:0]	Input	SDI data stream 8
SDI_DS_IN_ln_num_1[10:0] to SDI_DS_IN_ln_num_4	Input	SDI data stream line number
SDI_DS_IN_tx_ce	Input	Clock enable
SDI_DS_IN_tx_sd_ce	Input	SD-SDI mode clock enable
sdi_tx_err[31:0]	Output	Bit0: tx_ce_align_err; Bit31~bit1: Reserved;

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

Register Space

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

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

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 2-2.

Table 2-2: **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 AXI-Lite Interface** is selected in Vivado IDE.

The UHD-SDI TX IP core register space is shown below in Table 2-3.

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

Endianness

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

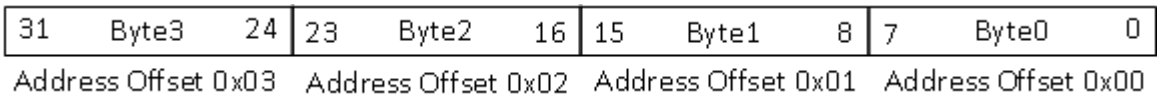

Figure 2-1: **32-bit Little Endian Example**

Table 2-3: **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

Table 2-3: **UHD-SDI TX IP Core Register Space *(Cont'd)***

Offset	Name	Width	Access	Description
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
0x44	RESERVED	32-bit	N/A	N/A
0x48	RESERVED	32-bit	N/A	N/A
0x4C	RESERVED	32-bit	N/A	N/A
0x50	RESERVED	32-bit	N/A	N/A
0x54	RESERVED	32-bit	N/A	N/A
0x58	RESERVED	32-bit	N/A	N/A
0x5C	RESERVED	32-bit	N/A	N/A
0x60	RESERVED	32-bit	N/A	N/A
0x64	RESERVED	32-bit	N/A	N/A
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

Notes:

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

RST_CTRL Register

The Core Control register (0x00 offset) is described in Table 2-4 and allows you to enable and disable the UHD-SDI TX IP core and apply a soft reset during core operation.

Table 2-4: **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 bit for AXI4S-to-Video out core 1 – AXI4S-to-Video out core is enabled 0 – AXI4S-to-Video out core is disabled
8	SDITX_BRIDGE_EN	R/W	0	Enable bit for SDI TX Bridge 1 – SDI TX Bridge is enabled 0 – SDI TX Bridge is disabled
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 SDI TX IP core will be resetted.
0	SDITX_SS_EN	R/W	0	Enable bit for SDI TX IP core 1 – SDI TX IP core is enabled 0 – SDI TX IP core is disabled

MODULE_CTRL Register

The Module Control register (0x04 offset) is described in Table 2-5 and allows you to control the UHD-SDI TX IP core and allows to change IP core functional modes.

Table 2-5: **MODULE_CTRL Register Bit Mapping**

Bits	Name	Access	Default Value	Description
31:21	Reserved	RO	0	Reserved
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.

Table 2-5: **MODULE_CTRL Register Bit Mapping *(Cont'd)***

Bits	Name	Access	Default Value	Description
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 11X 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.
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

Table 2-5: **MODULE_CTRL Register Bit Mapping *(Cont'd)***

Bits	Name	Access	Default Value	Description
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
3:0	Reserved	RO	0	Reserved

Global Interrupt Enable Register (GLBL_IER)

Global interrupt enable register (0x0C offset) is described in Table 2-6.

Table 2-6: **GLBL_IER Register Bit Mapping**

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

Interrupt Status Register (ISR)

The Interrupt Status register (0x10 offset) is described in Table 2-7 and captures the error and status information for the IP core.

Table 2-7: **ISR bit mapping**

Bits	Name	Access(1)	Default Value	Description
31:11	Reserved	RO	0	Reserved
10	UNDERFLOW_INTR	R/W1C	0	AXI-4 Stream to Video out core underflow indication
9	OVERFLOW_INTR	R/W1C	0	AXI-4 Stream to Video out core overflow indication
8	AXI4S_VID_LOCK_INTR	R/W1C	0	Lock indication from AXI-4 Stream to Video out core
7:2	Reserved	RO	0	Reserved
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, user has to write 1 to corresponding bits).

Interrupt Enable Register (IER)

The Interrupt Enable register (0x14 offset) is described in Table 2-8 and allows you to selectively generate an interrupt at the output port for each error/status bit in the ISR. An IER bit set to 0 does not inhibit an error/status condition from being captured, but inhibits it from generating an interrupt.

Table 2-8: **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 2-7.
10	UNDERFLOW_IE	R/W	0	
9	OVERFLOW_IE	R/W	0	
8	AXI4S_VID_LOCK_IE	R/W	0	
7:2	Reserved	RO	0	
1	TX_CE_ALIGN_ERR_IE	R/W	0	
0	GTTX_RSTDONE_IE	R/W	0	

TX_ST352_LINE Register

TX_ST352_LINE register (0x18 offset) is described in Table 2-9.

Table 2-9: **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

TX_ST352_DATA_DS1 register (0x1C offset) is described in Table 2-10.

Table 2-10: **TX_ST352_DATA_DS1 Register Bit Mapping**

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

TX_ST352_DATA_DS3 Register

TX_ST352_DATA_DS3 register (0x20 offset) is described in Table 2-11.

Table 2-11: **TX_ST352_DATA_DS3 Register Bit Mapping**

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

TX_ST352_DATA_DS5 Register

TX_ST352_DATA_DS5 register (0x24 offset) is described in Table 2-12.

Table 2-12: **TX_ST352_DATA_DS5 Register Bit Mapping**

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

TX_ST352_DATA_DS7 Register

TX_ST352_DATA_DS7 register (0x28 offset) is described in Table 2-13.

Table 2-13: **TX_ST352_DATA_DS7 Register Bit Mapping**

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

TX_ST352_DATA_DS9 Register

TX_ST352_DATA_DS9 register (0x2C offset) is described in Table 2-14.

Table 2-14: **TX_ST352_DATA_DS9 Register Bit Mapping**

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

TX_ST352_DATA_DS11 Register

TX_ST352_DATA_DS11 register (0x30 offset) is described in Table 2-15.

Table 2-15: **TX_ST352_DATA_DS11 Register Bit Mapping**

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

TX_ST352_DATA_DS13 Register

TX_ST352_DATA_DS13 register (0x34 offset) is described in Table 2-16.

Table 2-16: **TX_ST352_DATA_DS13 Register Bit Mapping**

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

TX_ST352_DATA_DS15 Register

TX_ST352_DATA_DS15 register (0x38 offset) is described in Table 2-17.

Table 2-17: **TX_ST352_DATA_DS15 Register Bit Mapping**

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

VERSION Register

VERSION register (0x3C offset) is described in Table 2-18.

Table 2-18: **VERSION Register Bit Mapping**

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

SS_CONFIG Register

SS_CONFIG register (0x40 offset) is described in Table 2-19.

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

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

SDI_TX_BRIDGE_STS Register

SDI_TX_BRIDGE_STS register (0x68 offset) is described in Table 2-20.

Table 2-20: **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;

Table 2-20: **SDI_TX_BRIDGE_STS Register Bit Mapping *(Cont'd)***

Bits	Name	Access	Default Value	Description
3:1	Reserved	RO	0	Reserved
0	SDITX_BRIDGE_SEL	RO	0	Select bit for SDI TX Bridge 0 – 3G SDI TX Bridge is selected 1 – 12G SDI TX Bridge is selected

AXI4S_VID_OUT_STS Register

AXI4S_VID_OUT_STS register (0x6C offset) is described in Table 2-21.

Table 2-21: **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

VTC Registers

The VTC registers are available when **Enable Vid-Over-AXIS Interface** is selected in Vivado IDE. For details about VTC registers, see the *Video Timing Controller Product Guide* [Ref 10] for details.

Designing with the Core

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

General Design Guidelines

This section describes the steps required to turn a SMPTE UHD-SDI 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 FPGA. This design can be used as a starting point for your own design or can be used to troubleshoot the user application, if necessary.

Know the Degree of Difficulty

SMPTE UHD-SDI 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

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. Frequency range could be 50 MHz to 150 MHz.
sdi_tx_clk	Core clock for UHD-SDI TX core. Refer Table 3-2 for more details.

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

Clock Name	Description
video_in_clk	Clock used for Video data conversion to SDI data stream. To support 12G-SDI for 10-bit YUV 4:2:2 in 2 PPC(2), clock must set to maximum 300MHz. 2*(BPC)(1)*(PPC)*clock = 2*10*2*300MHz = 12 Gbps The video_int_clk for 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 Bits per component that is set to 10 since Subsystem supports 10-bit YUV4:2:2.
2. PPC is Pixel Per Clock that is set to 2 by SDI bridge.

The frequency of sdi_TX_clk of UHD-SDI TX core is given Table 3-2.

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

SMPTE Standard	Supported Data Stream	Clock Frequency (in 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
3G-SDI Level B	4	148.5
6G-SDI	8	148.5
12G-SDI	8	297

See the *Clocking* section the *SMPTE UHD-SDI Product Guide* [Ref 8] for more clocking information.

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 UHD-SDI TX core and synchronous with `sdi_tx_clk`. Refer to the *Clocking* section the *SMPTE UHD-SDI Product Guide* [Ref 8].

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

Table 3-3: **Core Resets**

Sub-Core	s_axi_arstn	video_in_arstn	sdi_tx_rst
AXI-4 Stream to Video Out	NA	NA	NA
Video to SDI TX Bridge	NA	NA	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	NA	NA
AXI Crossbar	Connected to aresetn core port	NA	NA

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.

Design Flow Steps

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

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

Customizing and Generating the Core

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

Vivado Integrated Design Environment

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

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

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

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

Core Configuration Tab

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

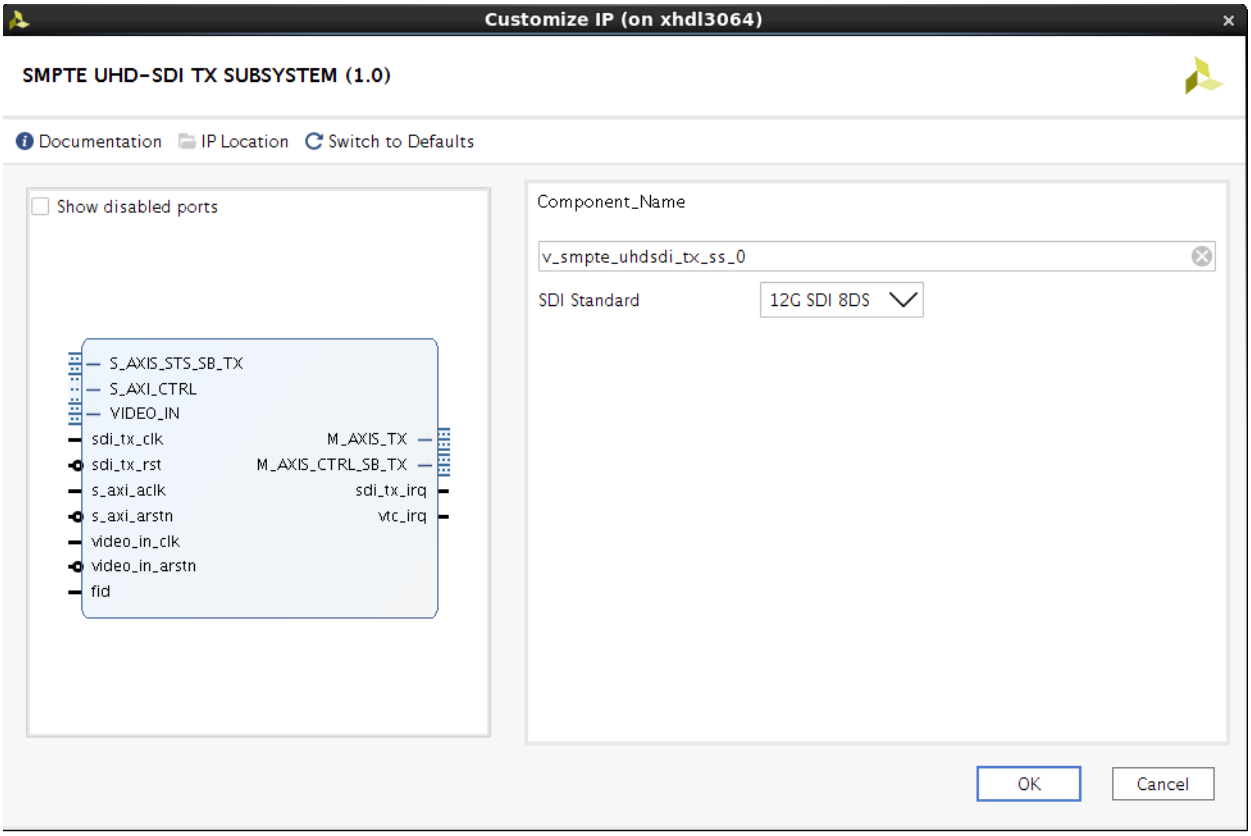

Figure 4-1: **Subsystem Configuration Tab**

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

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

User Parameters

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

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

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

Output Generation

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

Constraining the Core

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

Required Constraints

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

Device, Package, and Speed Grade Selections

This section is not applicable for this subsystem.

Clock Frequencies

See Clocking in Chapter 3.

Clock Management

The SMPTE UHD-SDI 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].

Synthesis and Implementation

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

Example Design

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

No Example design support is available for the SMPTE UHD-SDI TX Subsystem. However, example design for SMPTE UHD-SDI RX Subsystem is available. As part of the example design for SMPTE UHD-SDI RX Subsystem demonstration, the SMPTE UHD-SDI TX Subsystem is used for SDI data transfer. For more details on the options and features available refer to the *Example Design* section of the *SMPTE UHD-SDI RX Subsystem Product Guide* (PG290) [Ref 12].

Verification, Compliance, and Interoperability

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

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

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

Hardware Testing

The SMPTE UHD-SDI Transmitter 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.

Migrating and Upgrading

This appendix contains information about upgrading to a more recent version of the IP core.

Upgrading in the Vivado Design Suite

This section provides information about any changes to the user logic or port designations that take place when you upgrade to a more current version of this IP core in the Vivado Design Suite.

Debugging

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

Finding Help on Xilinx.com

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

Documentation

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

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

Solution Centers

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

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

- Xilinx Video Solution Center

Answer Records

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

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

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

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

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

Master Answer Record for the SMPTE UHD-SDI Transmitter Subsystem

AR: 68767

Technical Support

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

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

To contact Xilinx Technical Support, navigate to the Xilinx Support web page.

Debug Tools

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

Vivado Lab Edition

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

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

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

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

Hardware Debug

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 IO pin planning and XDC constraints.
- Follow recommended reset sequence.
- Verify all clocks are connected and are with expected frequencies.
- Enable AXI-4 Lite based register interface to get core status and control.
- Make sure serial line trace lengths are equal.
- Verify the FMC_VADJ voltage to 1.8v in case of FMC card usage.
- If your outputs go to 0, check your licensing.

Transceiver (GT) Clocking

- Make sure QPLL is getting reset before starting the IP.
- Monitor the QPLL LOCK signal.
- Verify that QPLL input clock frequency is of expected value.

  It is mandatory to reset the QPLL if clock input to QPLL is stopped or unstable. Refer the AR# 57738 for debugging GT reference clock issues.

- Make sure to use QPLL default settings from latest GT Wizard IP core based on target device.
- Check the voltage rails on the transceivers. Refer AR#57737 for more information.
- Measure TXOUTCLK is of expected frequency.

- Make sure TXOUTCLK of the transceiver 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 GT completes initialization.
- Make sure GT is not reset during normal operation.
- Please refer AR#59435 for more information on debugging GT reset problems.
- Follow recommended GT reset sequence.

Video Timing Controller (VTC) Debug:

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

AXI-Stream to Video Out Debug:

- Check 'locked' signal from AXI-Stream to Video Out core is asserted.
- Make sure overflow or underflow output signals are not asserted. If so, then check the connected clock frequencies and make sure that it matches with configured SDI mode line rate.
- Monitor status[31:0] to know about the status of AXI-Stream to Video Out core. This status bus is available in AXI4S_VID_OUT_STS register of AXI-4 Lite interface.

Video to SDI TX Bridge Debug:

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

UHD-SDI TX Core Debug:

- Make sure that UHD-SDI TX core is configured with 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.

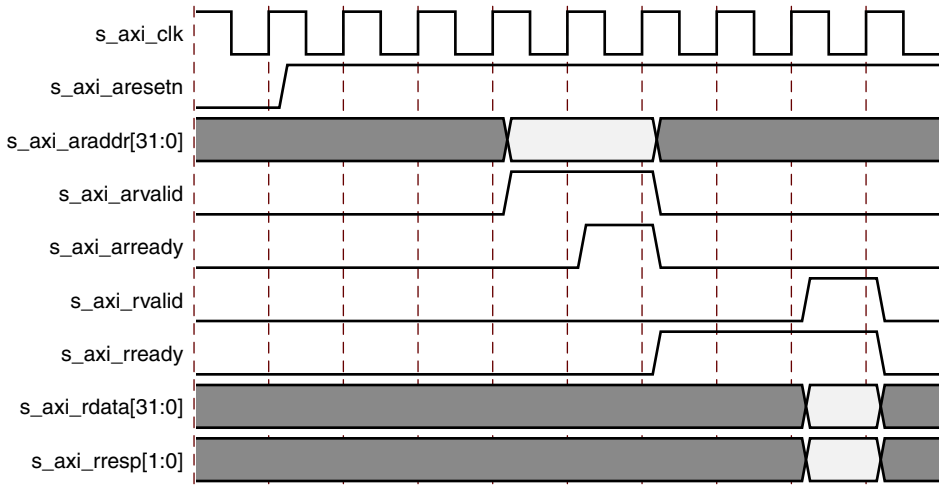

Figure C-1: **Read**

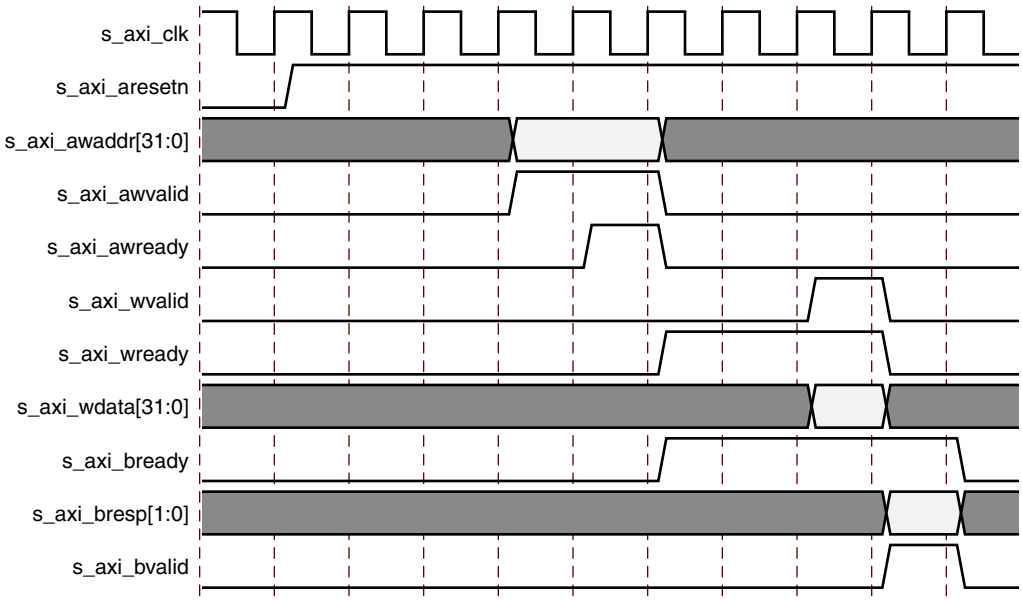

Figure C-2: **Write**

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. *ISE® to Vivado Design Suite Migration Guide* (UG911)
6. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
7. *Vivado Design Suite User Guide - Implementation* (UG904)
8. *SMPTE UHD-SDI Product Guide* (PG0205)
9. *AXI-4 Stream to Video Out Product Guide* (PG044)
10. *Video Timing Controller LogiCORE IP Product Guide* (PG016)
11. *AXI4-Stream Video IP and System Design Guide* (UG934)
12. *SMPTE UHD-SDI RX Subsystem Product Guide* (PG290)
13. *Vivado Design Suite: AXI Reference Guide* (UG1037)
14. *AXI Interconnect LogiCORE IP Product Guide* (PG059)

Revision History

The following table shows the revision history for this document.

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

Please Read: Important Legal Notices

of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

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