LogiCORE IP SMPTE 2022-5/6 Video over IP Transmitter v3.0

Product Guide for Vivado Design Suite

PG032 October 2, 2013

XILINX®

XILINX

Table of Contents

Introduction

The Xilinx LogiCORE™ IP SMPTE 2022-5/6 Video over IP Transmitter is a module for broadcast applications that requires bridging between SMPTE video connectivity standards (SD/HD/3G-SDI) and 10 Gb/s networks. It is capable of mapping SD/HD/3G-SDI video streams into Ethernet packets and adding systematically generated redundant data. This allows the receiver to detect and correct a limited number of packet errors without the need to ask the transmitter for retransmission of lost packets. The core is for developing Internet protocol-based systems to reduce overall cost in broadcast facility for distribution and routing of audio video data.

Features

- Encapsulate SD/HD/3G-SDI streams from up to 8 inputs (3 for the case of 3G-SDI) according to SMPTE 2022-6
- Per stream basis Forward Error Correction (FEC) in accordance to SMPTE 2022-5
- Supports Level A and Level B FEC operations
- Supports block-aligned and non block-aligned FEC operations
- Dynamic switching of L and D values in FEC matrix over AXI4-Lite interface
- Supports Virtual Local Area Network (VLAN)
- AXI4-Stream data interfaces
- AXI4-Lite control interface
- User configurable Ethernet, IP, User Datagram Protocol (UDP) and Real-time Transport Protocol (RTP) headers over AXI4-Lite interface
- Supports SD-SDI, HD-SDI, 3G-SDI Level-A, 3G-SDI Level-B single stream and 3G Level-B dual stream

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	Zynq®-7000 All Programmable SoC, Virtex®-7, Kintex®-7
Supported User Interfaces	AXI4-Lite, AXI4-Stream, AXI4
Resources	See Table 2-1, Table 2-2, and Table 2-3.
Provided with Core	
Design Files	Encrypted HDL
Example Design	*High Bit Rate Media Transport over IP Networks with Forward Error Correction* (XAPP590) [Ref 1]
Test Bench	Verilog and VHDL
Constraints File	XDC
Simulation Model	Encrypted RTL, VHDL Behavioral, VHDL or Verilog source HDL
Supported S/W Driver	N/A
Tested Design Flows[2]	
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. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Overview

As broadcast and communications markets converge and the use of IP networks for transport of video streams becomes more attractive to broadcasters and telecommunications companies alike, the adoption of 10 Gb/s Ethernet for the transmission of multiple uncompressed Serial Digital Interface (SDI) streams is becoming a major customer requirement. The industry is primarily looking at the SMPTE 2022 set of standards to create an open and interoperable way of connecting video over 10GbE equipment together and ensure that Quality of Service (QoS) is high and packet loss is kept to a minimum or recovered through FEC. As shown in Figure 1-1, high bit rate SMPTE 2022-5/6 is aimed at contribution networks (for example, between broadcast center and regional studio).

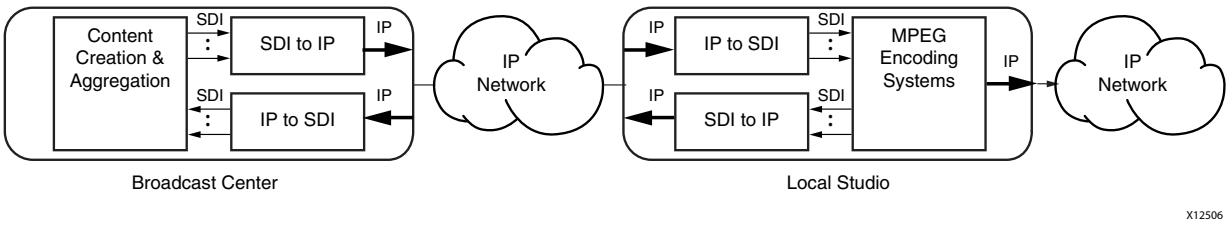

Figure 1-1: **High Bit Rate SMPTE 2022-5/6 between Broadcast Center and Local Studio**

The core includes Forward Error Correction (FEC). FEC protects the video stream during transport of high-quality video over IP networks. With FEC, the transmitter adds systematically generated redundant data to its video. This carefully designed redundancy allows the receiver to detect and correct a limited number of packet errors occurring anywhere in the video without the need to ask the transmitter for additional video data.

These errors, in the form of lost video packets, can be caused by many reasons, from thermal noise to storage system defects and transmission noise introduced by the environment. FEC gives the receiver the ability to correct these errors without needing a reverse channel to request retransmission of data. In real time systems, the latency is too great to request a retransmission. The ability of Xilinx FPGAs to bridge the broadcast and the communications industries by performing highly integrated real-time video interfaces helps broadcasters reduce costs as well as reduce the overall time it takes to acquire, edit and produce content. Now that video can be reliably delivered over 10 Gb/s Ethernet (10GbE), broadcasters can replace some of the expensive mobile infrastructures supporting outside live broadcasts, as well as enabling remote production from existing fixed studio set ups, dramatically reducing both capital expenditure and operating expenses.

Feature Summary

The core maps raw SD/HD/3G-SDI video streams into Ethernet packets as per SMPTE 2022-6. For each media stream with SMPTE 2022-6, the core creates the Forward Error Correction streams in accordance with SMPTE 2022-5 for recovery of IP packets lost to network transmission errors and ensure the highest picture quality of uncompressed, high bandwidth professional video.

The core support of VLAN comes from being able to operate seamlessly when receiving VLAN tagged Ethernet packets. You can configure and instantiate the core from the Vivado® Design Suite. Core functionality can be controlled dynamically through an AXI4-Lite interface.

Applications

- Transport uncompressed high bandwidth professional video streams over IP networks.
- Support real-time audio/video applications such as contribution, primary distribution, and digital cinema

Licensing and Ordering Information

License Checkers

If the IP requires a license key, the key must be verified. The Vivado design tools have several license check points for gating licensed IP through the flow. If the SMPTE20222-5/6 TX Core license check succeeds, the IP can continue generation. Otherwise, generation halts with error. License checkpoints are enforced by the following Vivado flow:

Vivado flow: Vivado Synthesis, Vivado Implementation, write_bitstream (Tcl Console command)

IMPORTANT: *IP license level is ignored at checkpoints. The test confirms a valid license exists. It does not check IP license level.*

If a Hardware Evaluation License is being used, the core will stop transmitting Ethernet packets after timeout.

License Type

This Xilinx LogiCORE™ IP module is provided under the terms of the Xilinx Core License Agreement. The module is shipped as part of the Vivado Design Suite. For full access to all core functionalities in simulation and in hardware, you must purchase a license for the core. Contact your local Xilinx sales representative for information about pricing and availability.

For more information, visit the SMPTE 2022-5/6 Video Over IP product web page.

Information about other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Product Specification

Standards

The core is compliant with the AXI4, AXI4-Stream and AXI4-Lite interconnect standards. See the "Video IP: AXI Feature Adoption" section of the *AXI Reference Guide* (UG761) [Ref 2] for additional information. The function of the core is compliant with *SMPTE 2022-5/6* standard.

Maximum Frequencies

The maximum achievable clock frequency can vary. The maximum achievable clock frequency and all resource counts can be affected by other tool options, additional logic in the FPGA, using a different version of Xilinx tools and other factors. See the resource utilization tables for device family specific information.

Resource Utilization

Resources required for this core have been estimated for Zynq®-7000 All Programmable SoC, Virtex®-7, and Kintex®-7 devices. These values were generated using Xilinx Vivado® Design Suite.

Table 2-1: **Resource Utilization for Zynq-7000 Devices (xc7z045, speed -1)**

SDI CHANNEL	FEC INCLUDE	FFs	LUTs	Slices	LUT FF Pairs	36k BLock RAMs	18k BLock RAMs	DSP48E1s
1	0	9,494	7,672	2,985	9,760	16	2	0
2	0	12,518	10,154	4,031	12,798	18	2	0
3	0	15,670	11,565	5,079	15,990	20	2	0
4	0	18,833	13,375	6,260	19,135	22	2	0
5	0	21,992	15,775	6,977	22,141	31	2	0
6	0	25,149	17,391	7,605	24,688	33	2	0
7	0	28,047	18,496	8,154	26,849	35	2	0
8	0	31,268	19,506	9,710	31,153	37	2	0
1	1	13,067	9,948	4,203	13,391	52	7	0
2	1	16,567	13,099	5,031	1,6746	54	7	0
3	1	20,105	15,123	6,586	20,566	56	7	0
4	1	23,585	16,755	7,258	23,467	58	7	0
5	1	27,120	19,562	8,009	26,697	67	7	0
6	1	30,630	21,926	9,320	30,408	69	7	0
7	1	33,881	23,339	10,659	34,160	71	7	0
8	1	37,487	24,750	12,115	38,365	73	7	0

Table 2-2: **Resource Utilization for Virtex 7 FPGAs (xc7vx690t, speed -1)**

SDI CHANNEL	FEC INCLUDE	FFs	LUTs	Slices	LUT FF Pairs	36k BLock RAMs	18k BLock RAMs	DSP48E1s
1	0	9,478	7,657	2,989	9,830	16	2	0
2	0	12,518	10,149	4,050	12,880	18	2	0
3	0	15,654	11,549	4,852	15,764	20	2	0
4	0	18,817	13,362	6,442	19,354	22	2	0
5	0	21,976	15,779	7,187	22,308	31	2	0
6	0	25,149	17,401	8,448	25,555	33	2	0
7	0	28,047	18,493	8,765	27,738	35	2	0
8	0	31,284	19,368	8,990	29,942	37	2	0
1	1	13,083	9,946	4,179	13,385	52	7	0
2	1	16,551	13,160	5,625	17,184	54	7	0
3	1	20,089	15,087	6,685	20,774	56	7	0
4	1	23,585	16,758	7,883	23,998	58	7	0
5	1	27,136	19,567	9,407	28,077	67	7	0
6	1	30,630	21,916	10,155	31,484	69	7	0
7	1	33,881	23,330	10,406	33,656	71	7	0
8	1	37,471	24,735	11,557	37,416	73	7	0

Table 2-3: **Resource Utilization for Kintex-7 FPGAs (xc7k325t, speed -1)**

SDI CHANNEL	FEC INCLUDE	FFs	LUTs	Slices	LUT FF Pairs	36k BLock RAMs	18k BLock RAMs	DSP48E1s
1	0	9,494	7,662	3,233	10,087	16	2	0
2	0	12,518	10,148	3,954	12,814	18	2	0
3	0	15,654	11,556	5,132	16,018	20	2	0
4	0	18,833	13,358	5,918	18,799	22	2	0
5	0	21,992	15,788	7,461	22,507	31	2	0
6	0	25,133	17,440	8,566	25,713	33	2	0
7	0	28,047	18,495	8,476	27,065	35	2	0
8	0	31,268	19,500	9,036	30,235	37	2	0
1	1	13,067	9,936	3,870	13,023	52	7	0
2	1	16,567	13,093	5,125	16,773	54	7	0
3	1	20,105	15,091	5,817	19,887	56	7	0
4	1	23,585	16,769	7,563	23,859	58	7	0
5	1	27,120	19,563	8,341	27,131	67	7	0
6	1	30,646	21,979	9,642	30,921	69	7	0
7	1	33,881	23,333	11,242	34,881	71	7	0
8	1	37,487	24,741	11,558	37,333	73	7	0

Port Descriptions

The core uses industry standard control and data interfaces to connect to other system components. The following sections describe the various interfaces available with the core. Figure 2-1 shows an I/O Diagram of the core. The RX_SDI interface pins depend on the number of channels configured through the Vivado Integrated Design Environment (IDE).

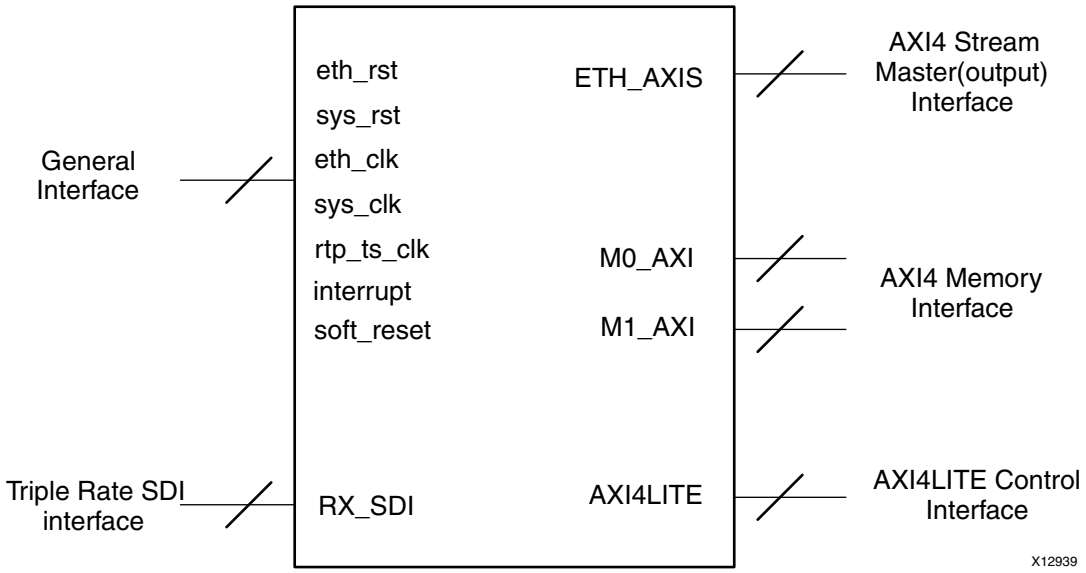

Figure 2-1: **Core Top Level Signaling Interface**

General Interface

Table 2-4 summarizes the signals which are either shared by, or are not part of the dedicated SDI, AXI4-Stream, AXI4 or AXI4-Lite control interfaces.

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

Signal Name	Direction	Width	Description
eth_rst	In	1	Ethernet domain reset
eth_clk	In	1	156.25MHz Ethernet clock
sys_rst	In	1	System domain reset
sys_clk	In	1	200MHz system clock.
rtp_ts_clk	In	1	27MHz RTP timestamp clock.
Interrupt	Out	1	Interrupt from processor
Soft_reset	Out	1	Reset from processor

AXI Memory Interface

The core uses an AXI4 interface to connect to the AXI4 interconnects. The AXI4 Interconnect provides the access to the external memory through AXI Double Data Rate (DDR) controller. See the *LogiCORE IP AXI Interconnect Product Guide* (PG059) [Ref 3] for more information.

Table 2-5: **AXI4 Memory Interface Signals**

Signal Name	Direction	Width	Description
m0_axi_awid	Out	1	Write Address Channel Transaction ID
m0_axi_awaddr	Out	32	Write Address Channel Address
m0_axi_awlen	Out	8	Write Address Channel Burst Length code
m0_axi_awsize	Out	3	Write Address Channel Transfer Size code
m0_axi_awburst	Out	2	Write Address Channel Burst Type
m0_axi_awlock	Out	2	Write Address Channel Atomic Access Type
m0_axi_awcache	Out	4	Write Address Channel Cache Characteristics
m0_axi_awport	Out	3	Write Address Channel Protection Bits
m0_axi_awqos	Out	4	Write Address Channel Quality of Service
m0_axi_awvalid	Out	1	Write Address Channel Valid
m0_axi_awready	In	1	Write Address Channel Ready
m0_axi_wdata	Out	256	Write Data Channel Data
m0_axi_wstrb	Out	32	Write Data Channel Data Byte Strobes
m0_axi_wlast	Out	1	Write Data Channel Last Data Beat
m0_axi_wvalid	Out	1	Write Data Channel Valid
m0_axi_wready	In	1	Write Data Channel Ready
m0_axi_bid	In	1	Write Response Channel Transaction ID
m0_axi_bresp	In	2	Write Response Channel Response Code
m0_axi_bvalid	In	1	Write Response Channel Valid
m0_axis_bready	Out	1	Write Response Channel Ready
m0_axi_arid	Out	1	Read Address Channel Transaction ID
m0_axi_araddr	Out	32	Read Address Channel Address
m0_axi_arlen	Out	8	Read Address Channel Burst Length code
m0_axi_arsize	Out	3	Read Address Channel Transfer Size code
m0_axi_arburst	Out	2	Read Address Channel Burst Type
m0_axi_arlock	Out	2	Read Address Channel Atomic Access Type

Table 2-5: **AXI4 Memory Interface Signals *(Cont'd)***

Signal Name	Direction	Width	Description
m0_axi_arcache	Out	4	Read Address Channel Cache Characteristics
m0_axi_arprot	Out	3	Read Address Channel Protection Bits
m0_axi_arqos	Out	4	AXI4 Read Address Channel Quality of Service
m0_axi_arvalid	Out	1	Read Address Channel Valid
m0_axi_arready	In	1	Read Address Channel Ready
m0_axi_rid	In	1	Read Data Channel Transaction ID
m0_axi_rdata	In	256	Read Data Channel Data
m0_axi_rresp	In	2	Read Data Channel Response Code
m0_axi_rlast	In	1	Read Data Channel Last Data Beat
m0_axi_rvalid	In	1	Read Data Channel Valid
m0_axi_rready	Out	1	Read Data Channel Ready
m1_axi_arid	Out	1	Read Address Channel Transaction ID
m1_axi_araddr	Out	32	Read Address Channel Address
m1_axi_arlen	Out	8	Read Address Channel Burst Length code
m1_axi_arsize	Out	3	Read Address Channel Transfer Size code
m1_axi_arburst	Out	2	Read Address Channel Burst Type
m1_axi_arlock	Out	2	Read Address Channel Atomic Access Type
m1_axi_arcache	Out	4	Read Address Channel Cache Characteristics
m1_axi_arprot	Out	3	Read Address Channel Protection Bits
m1_axi_arqos	Out	4	AXI4 Read Address Channel Quality of Service
m1_axi_arvalid	In	1	Read Address Channel Valid
m1_axi_arready	In	1	Read Address Channel Ready
m1_axi_rid	In	1	Read Data Channel Transaction ID
m1_axi_rdata	In	256	Read Data Channel Data
m1_axi_rresp	In	2	Read Data Channel Response Code
m1_axi_rlast	In	1	Read Data Channel Last Data Beat
m1_axi_rvalid	In	1	Read Data Channel Valid
m1_axi_rready	Out	1	Read Data Channel Ready

AXI4-Stream Master Interface: Transmit

See the *LogiCORE IP 10-Gigabit Ethernet MAC Product Guide* (PG072) [Ref 4] for more information.

Table 2-6: **AXI4-Stream Interface Signals**

Signal Name	Direction	Width	Description
m_axis_aresetn	Out	1	AXI4-Stream Active-Low reset for Transmit path XGMAC.
m_axis_tdata[63:0]	Out	64	AXI4-Stream Data to XGMAC
m_axis_tkeep[7:0]	Out	8	AXI4-Stream Data Control to XGMAC.
m_axis_tvalid	Out	1	AXI4-Stream Data Valid input to XGMAC.
m_axis_tlast	Out	1	AXI4-Stream last Data input to XGMAC.
m_axis_tready	In	1	AXI4-Stream acknowledges signals from XGMAC to indicate to start the data transfer.

Society of Motion Picture and Television Engineers (SMPTE) SD/HD/3G-SDI 2.0 Interface

See the *Society of Motion Picture and Television Engineers (SMPTE) SD/HD/3G-SDI 2.0 Product Guide* (PG071) [Ref 5] for more information.

Table 2-7: **SMPTE SD/HD/3G-SDI Interface Signals**

Signal Name	Direction	Width	Description
rx_rst	In	1	Reset
rx_clk	In	1	Connect to rx_usrclk of SMPTE SD/HD/3G-SDI core.
rx_mode_locked	In	1	Connect to rx_mode_locked of SMPTE SD/HD/3G-SDI core.
rx_locked	In	1	Connect to rx_t_locked of SMPTE SD/HD/3G-SDI core.
rx_t_family	In	4	Connect to rx_t_family of SMPTE SD/HD/3G-SDI core.
rx_t_rate	In	4	Connect to rx_t_tate of SMPTE SD/HD/3G-SDI core.
rx_bit_rate	In	1	Connect to rx_bit_rate of SMPTE SD/HD/3G-SDI core.
rx_mode	In	2	Connect to rx_mode of SMPTE SD/HD/3G-SDI core.
rx_eav	In	1	Connect to rx_eav of SMPTE SD/HD/3G-SDI core.
rx_ce_sd	In	1	Connect to rx_ce_sd of SMPTE SD/HD/3G-SDI core.
rx_dout_rdy_3g	In	1	Connect to rx_dout_rdy_3g of SMPTE SD/HD/3G-SDI core.
rx_crc_err_a	In	1	Connect to rx_crc_err_a of SMPTE SD/HD/3G-SDI core.

Table 2-7: **SMPTE SD/HD/3G-SDI Interface Signals *(Cont'd)***

Signal Name	Direction	Width	Description
rx_a_vpid_valid	In	1	Connect to rx_a_vpid_valid of SMPTE SD/HD/3G-SDI core.
rx_a_vpid	In	32	Connect to rx_a_vpid of SMPTE SD/HD/3G-SDI core.
rx_line_a	In	11	Connect to rx_line_a of SMPTE SD/HD/3G-SDI core.
rx_ds1_a	In	10	Connect to rx_ds1a of SMPTE SD/HD/3G-SDI core.
rx_ds2_a	In	10	Connect to rx_ds2a of SMPTE SD/HD/3G-SDI core.
rx_ds1_b	In	10	Connect to rx_ds1b of SMPTE SD/HD/3G-SDI core.
rx_ds2_b	In	10	Connect to rx_ds2b of SMPTE SD/HD/3G-SDI core.
rx_level_b_3g	In	1	Connect to rx_level_b_3g of SMPTE SD/HD/3G-SDI core.

AXI4-Lite Control Interface

The AXI4-Lite interface allows you to dynamically control parameters within the core. Core configuration can be accomplished using an embedded ARM® or soft system processor such as MicroBlaze™.

The core can be controlled through the AXI4-Lite interface using read and write transactions to the SMPTE 2022-5/6 Video over IP Transmitter register space.

The AXI4-Lite slave interface facilitates integrating the core into a processor system, or along with other video or AXI4-Lite compliant IP, connected through the AXI4-Lite interface to an AXI4-Lite master.

Table 2-8: **AXI4-Lite Interface Signals**

Signal Name	Direction	Width	Description
s_axi_clk	In	1	Clock
s_axi_aresetn	In	1	AXI4-Lite Active-Low reset
s_axi_awaddr	In	9	AXI4-Lite Write Address Bus
s_axi_awvalid	In	1	AXI4-Lite Write Address Channel Write Address Valid
s_axi_wdata	In	32	AXI4-Lite Write Data Bus
s_axi_wstrb	In	4	AXI4-Lite Write Data Channel Data Byte Strobes
s_axi_wvalid	In	1	AXI4-Lite Write Data Channel Write Data Valid
s_axi_awready	Out	1	AXI4-Lite Write Address Channel Write Address Ready. Indicates DMA ready to accept the write address.
s_axi_wready	Out	1	AXI4-Lite Write Data Channel Write Data Ready. Indicates DMA is ready to accept the write data.

Table 2-8: **AXI4-Lite Interface Signals *(Cont'd)***

Signal Name	Direction	Width	Description
s_axi_bresp	Out	2	AXI4-Lite Write Response Channel. Indicates results of the write transfer.
s_axi_bvalid	Out	1	AXI4-Lite Write Response Channel Response Valid. Indicates response is valid.
s_axi_bready	In	1	AXI4-Lite Write Response Channel Ready. Indicates target is ready to receive response.
s_axi_arvalid	In	1	AXI4-Lite Read Address Channel Read Address Valid
s_axi_arready	Out	1	Ready. Indicates DMA is ready to accept the read address.
s_axi_araddr	In	9	AXI4-Lite Read Address Bus
s_axi_rready	In	1	AXI4-Lite Read Data Channel Read Data Ready. Indicates target is ready to accept the read data.
s_axi_rdata	Out	32	AXI4-Lite Read Data Bus
s_axi_rresp	Out	2	AXI4-Lite Read Response Channel Response. Indicates results of the read transfer.
s_axi_rvalid	Out	1	AXI4-Lite Read Data Channel Read Data Valid

Register Space

The core register space is partitioned to General and Channel specific registers. See the *SMPTE 2022-5/6* reference design for more information on register usage.

Table 2-9: **AXI4-Lite Register Map**

Address (hex) BASEADDR +	Register Name	Access Type	Default Value	Register Description
General Registers				
0x0000	CONTROL	R/W	0	Bit 0: Reserved Bit 1: REG_UPDATE 31 - 2: Reserved
0x0004	RESET	R/W	0	Bit 0: RESET 31-1: Reserved
0x0030	CHANNEL	R/W	0	31-0: Channel to access
0x003C	VERSION	R	0x03000000	7-0: REVISION_NUMBER 11-8: PATCH_ID 15-12: VERSION_REVISION 23-16: VERSION_MINOR 31-24:VERSION_MAJOR
0x0050	AXI_MM_ADDDR_MSB	R/W	0	2-0: Most significant 3 bits of the 32-bit AXI memory map address to access the DDR through the AXI interconnect 31-3: Reserved
0x0060	SRC_MAC_ADDR_LOW	R/W	0	31-0: Source Media Access Controller (MAC) address [31:0]
0x0064	SRC_MAC_ADDR_HIGH	R/W	0	15-0: Source MAC address [47:32] 31-16: Reserved
0x0068	SRC_IP_ADDR	R/W	0	31-0: Source IP address
0x008C	VLAN	R/W	0	11-0: VID Bit 12: CFI 15-13: Priority 30-16: Reserved Bit 31: VLAN enable
0x0084	HDR_PARAM	R/W	0	Bit 0: Reserved Bit 1: Include Video Timestamp 31-2: Reserved

Table 2-9: **AXI4-Lite Register Map *(Cont'd)***

Address (hex) BASEADDR +	Register Name	Access Type	Default Value	Register Description
0x00A0	NUM_CHANNEL	R	0	10-0: Number of channels in design 31-11: reserved
Channel Registers				
0x0100	CHAN_EN	R/W	0	Bit 0: Channel Enable 31-1: Reserved
0x0104	FEC_CONFIG	R/W	0	Bit 0: Reserved Bit 1: Row FEC enable Bit 2: Column FEC enable 31-3: Reserved
0x0108	FEC_OFFSET	R/W	0	Bit 0: Non block-aligned On/Off 31-1: Reserved
0x010C	FEC_L	R/W	0	9-0: FEC_L 31-10: Reserved
0x0110	FEC_D	R/W	0	9-0: FEC_D 31-10: Reserved
0x0120	DEST_MAC_ADDR_LOW	R/W	0	31-0: Destination MAC address [31:0]
0x0124	DEST_MAC_ADDR_HIGH	R/W	0	31-16: Reserved 15-0: Destination MAC address [47:32]
0x0128	DEST_IP_ADDR	R/W	0	31-0: Destination IP address
0x0138	SRC_UDP_PORT	R/W	0	31-16: Reserved 15-0: Source UDP port
0x013C	DEST_UDP_PORT	R/W	0	31-16: Reserved 15-0: Destination UDP port
0x0140	SSRC	R/W	0	31-0: Synchronization Source (SSRC)
0x0148	VID_LOCK_PARAM	R	0	Bit 0: Video Locked 2-1: SDI format 31-3: Reserved

CONTROL (0x0000) Register

Bit 1 of the CONTROL register is a write-done semaphore for the host processor, which facilitates committing all user register updates in the channel space simultaneously. One set of registers (the processor registers) is directly accessed by the processor interface, while the other set (the active set) is actively used by the core.

New values written to the processor registers are copied over to the active set if and only if the register update bit is set. Setting the bit to 0 before updating multiple registers and then setting the bit to 1 when updates are completed ensures all channel space registers are updated simultaneously.

RESET (0x0004) Register

Bit 0 facilitates software reset. When 1, all registers and the core are held at reset.

CHANNEL (0x0030) Register

Bit fields of this register set the channel number to read and write to/from registers in the channel space. All the channels share the same set of register address in the channel space.

Version (0x003C) Register

Bit fields of the Version Register facilitate software identification of the exact version of the hardware peripheral incorporated into a system. The core driver can take advantage of this Read-Only value to verify that the software is matched to the correct version of the hardware.

AXI_MM_ADDR_MSB (0x0050) Register

This register configures the most significant three bits of the 32-bit AXI memory map address to access the DDR through the AXI interconnect.

SRC_MAC_ADDR_LOW (0x0060) Register

This register configures the third, fourth, fifth and sixth bytes of the source Ethernet MAC Address that is inserted into the Ethernet header of the packet.

SRC_MAC_ADDR_HIGH (0x0064) Register

This register configures the first byte and second byte of the source Ethernet MAC Address that is inserted into the Ethernet header of the packet.

SRC_IP_ADDR (0x0068) Register

This register configures the source IP address that is inserted into the IP header of each packet.

HDR_PARAM (0x0084) Register

To include video timestamp in SMPTE 2022-6 header, set bit 1 of this register to '1'.

VLAN (0x008C) Register

VLAN register configures whether the Ethernet packet contains a VLAN tag and the tag control information to insert into each packet. The Tag Protocol Identifier is set to 0x8100.

NUM_CHANNEL (0x00A0) Register

This register indicates the number of channels in the design.

CHAN_EN (0x0100) Register

This register enables the channel to work by setting to '1'.

FEC_CONFIG (0x0104) Register

Bit 1 and Bit 2 of this register are for configuring the forward error correction level. For level B FEC sets both bits and for level A FEC sets Bit 2.

FEC_OFFSET (0x0108) Register

The FEC_OFFSET register turns on/off the non block-aligned feature of the FEC engine.

FEC_L (0x010C) Register

The FEC_L register configures the L value of the FEC matrix.

- Level A FEC 1 <= L<= 1020 and
- Level B FEC 4 <= L <= 1020

FEC_D (0x0110) Register

The FEC_D register configures the D value of the FEC matrix.

Both level A and level B FEC 4 <= D <= 255.

L x D shall be <= 1500 in SD, <= 3000 for HD (1.485 Gb/s) <= 6000 for 3G HD.

DEST_MAC_ADDR_LOW (0x0120) Register

This register configures the third, fourth, fifth and sixth bytes of the destination Ethernet MAC Address that is inserted into the Ethernet header of the packet.

DEST_MAC_ADDR_HIGH (0x0124) Register

This register configures the first byte and second byte of the destination Ethernet MAC Address that is inserted into the Ethernet header of the packet.

DEST_IP_ADDR (0x0128) Register

This register configures the destination IP address that is inserted into the IP header of each packet.

SRC_UDP_PORT (0x0138) Register

This register configures the UDP source port value that is inserted into the UDP header of each packet.

DEST_UDP_PORT (0x013C) Register

This register configures the UDP destination port value that is inserted into the UDP header of each packet.

SSRC (0x0140) Register

This register configures the SSRC value that is inserted into the RTP header of each packet.

VID_LOCKED_PARAM (0x0148) Register

Bit 0 is High when the input video from SMPTE SD/HD/3G-SDI is locked to a video format. Bit 2 and bit 3 indicate the current SDI mode of the channel.

- 0 0= HD-SDI
- 0 1= SD-SDI
- 1 0= 3G-SDI

Designing with the Core

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

The core is for broadcast applications that require bridging between SMPTE video connectivity standards SD/HD/3G-SDI and 10 Gb/s Ethernet. The core takes uncompressed SD/HD/3G-SDI streams as input from the SMPTE SD/HD/3G-SDI core, encapsulates the data using prescribed methods into an IP packet with UDP and RTP header together with Forward Error Correction in accordance with SMPTE 2022-5/6, and sends over the AXI4-Stream interface to the 10G Ethernet MAC. The core uses AXI4 interface to transfer data between the core and buffer in external DDR memory. The register interface is compliant with AXI4-Lite interface. See the *SMPTE2022-5/6 High Bit Rate Media Transport over IP Networks with Forward Error Correction on Kintex-7 FPGAs* (XAPP896) [Ref 6] reference design for more information.

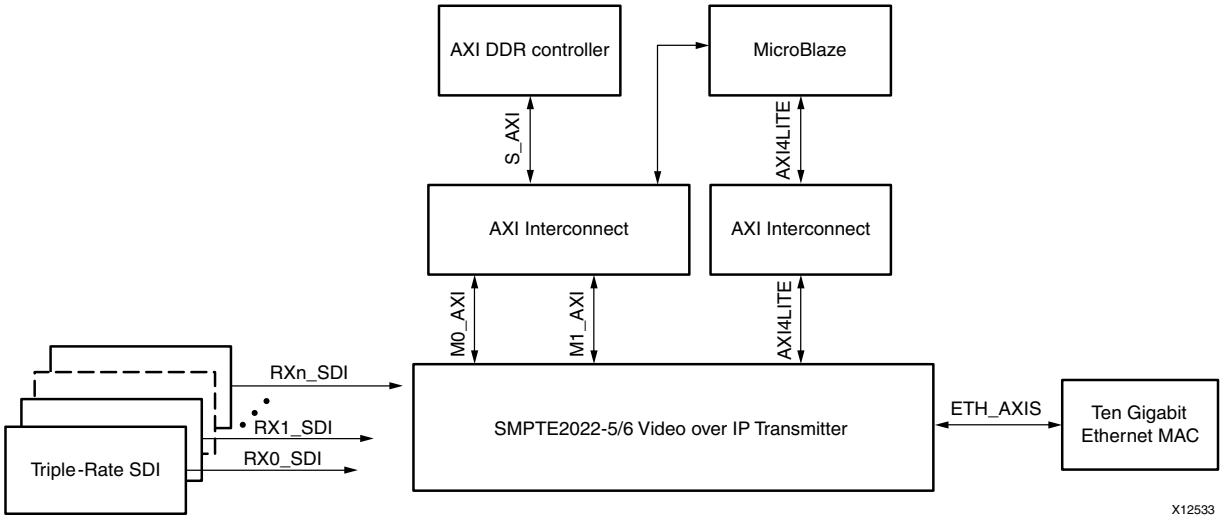

Figure 3-1: **SMPTE 2022-5/6 Video over IP Transmitter System Built with other Xilinx IP Cores**

Note: There is an option to include Forward Error Correction engine in the SMPTE 2022-5/6 Video over IP Transmitter core. Adding this enables the receiver to recover IP packets lost to the network transmission errors and hence ensure the quality of the uncompressed video. However, it will increase the resource count in the FPGA as well as the usage of external memory.

Clocking

The core has four clock domains.

- SDI video clock domain
- System clock domain recommended running at 200 MHz
- Ethernet clock domain at 156.25 MHz
- AXI4-Lite clock domain recommended at 100 MHz)

Resets

See the *SMPTE2022-5/6 High Bit Rate Media Transport over IP Networks with Forward Error Correction on Kintex-7 FPGAs* (XAPP896) [Ref 6] reference design for more information.

Memory Requirement

Table 3-1 shows a tabulation of the amount of DDR memory required by the SMPTE 2022-5/6 Video over IP Transmitter core based on the number of channels instantiated in the design when the Forward Error Correction engine is included.

Table 3-1: **Memory Requirement for the SMPTE 2022-5/6 Video over IP Transmitter Core with Forward Error Correction Engine**

Number of Channels Instantiated	Size of DDR Memory Needed (MB)
1	4
2	8
3	12
4	16
5	20
6	24
7	28
8	32

Customizing and Generating the Core

This chapter includes information about using Xilinx tools to customize and generate the core 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 core using the following steps:

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

For details, see the sections, "Working with IP" and "Customizing IP for the Design" in the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 7] and the "Working with the Vivado IDE" section in the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 8]

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

Vivado Integrated Design Environment (IDE)

The core is configured to meet the specific needs of the developer before instantiation through the Vivado IDE. This section provides a quick reference to parameters that can be configured at generation time.

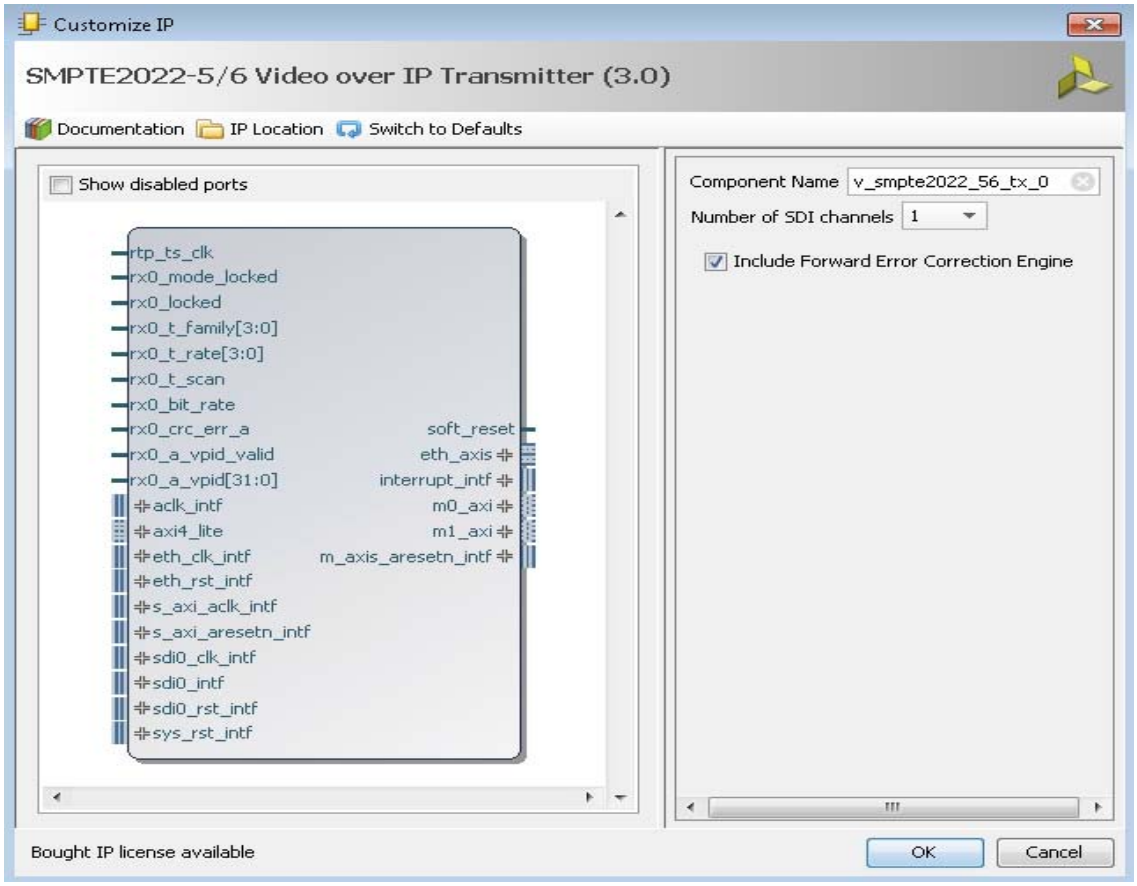

Figure 4-1: **Vivado IDE**

The Vivado IDE shows a representation of the IP symbol on the left side, and the parameter assignments on the right side, which are described as follows:

- **Component Name:** The component name is used as the base name of output files generated for the module. Names must begin with a letter and must be composed from characters: a to z, 0 to 9 and "_". The name v_smpte2022_56_tx cannot be used as a component name.
- **Number of SDI Channels:** Specifies the number of SDI channels.
- **Include Forward Error Correction engine**: When this option is checked, the core is generated with Forward Error Correction Engine.

Output Generation

The Vivado design tools generate the files necessary to build the core and places those files in the <project>/<project>.srcs/sources_1/ip/<core> directory.

For details, see "Generating IP Output Products" in the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 7].

Constraining the Core

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

Required Constraints

Constraints required for the core are clock frequency constraints for the clock domains described in Clocking in Chapter 3. Paths between the clock domains are constrained with a max_delay constraint and use the datapathonly flag, causing setup and hold checks to be ignored for signals that cross clock domains. These constraints are provided in the XDC constraints file included with the core.

Device, Package, and Speed Grade Selections

There are no device, package or speed grade requirements for this core. This core has not been characterized for use in low-power devices.

Clock Frequencies

See Maximum Frequencies in Chapter 2.

Clock Management

See Clocking in Chapter 3.

Clock Placement

There are no specific clock placement requirements for this core.

Banking

There are no specific banking rules for this core.

Transceiver Placement

There are no transceiver placement requirements for this core.

I/O Standard and Placement

There are no specific I/O standards and placement requirements for this core.

Synthesis and Implementation

For details about synthesis and implementation, see "Synthesizing IP" and "Implementing IP" in the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 7].

Test Bench

This chapter contains information about the provided test bench in the Vivado® Design Suite.

Demonstration Test Bench

A demonstration test bench is provided with the core that enables you to observe core behavior in a typical scenario. This test bench is generated together with the core in the Vivado Design Suite. You are encouraged to make simple modifications to the configurations and observe the changes in the waveform.

Directory and File Contents

The following files are expected to be generated in the in the demonstration test bench output directory.

- ddr_ram_dummy.vhd
- axi4_rd_handler.vhd
- axi4_wr_handler.vhd
- axi4_dummy_tx_memory.vhd
- axi4_dummy_rx_memory.vhd
- trsgen.vhd
- sof_detect_stream.vhd
- axi4lite_mst.v
- ram_dp_ar_aw.v
- syn_fifo.v
- sdi_stream_data_checker.v
- sdi_video_gen.v
- tb_<IP_instance_name>.v

Note: The VOIP Transmitter Core uses an encrypted version of the VOIP Receiver core in a loopback mode in the test bench. You cannot view the encrypted module.

Test Bench Structure

The top-level entity is tb_<IP_instance_name>. It instantiates the following modules.

- UUT
  The <IP> core instance under test.
- VIDGEN
  The SDI Input Video generator module that feeds input data [SD/HD/3G-SDI] to the VOIP Transmitter Core.
- i_DDR_VOIP_TX
  The DDR dummy memory model instance using the AXI4 memory interface to emulate data transfer between the transmitter core and external DDR memory
- AXI4LITE_MST_TX
  The AXI4-Lite master module, which initiates AXI4-Lite transactions to program VOIP Transmitter core registers.
- i_DDR_VOIP_RX
  The DDR dummy memory model instance using the AXI4 memory interface to emulate data transfer between the Receiver core and external DDR memory.
- AXI4LITE_MST
  The AXI4-Lite master module initiates AXI4-Lite transactions to program VOIP Receiver core registers.
- STREAM_CHECKER
  The end-to-end stream data checker module instance to check the data integrity of the Stream Input/Output Data from the VOIP Transmitter core to the VOIP Receiver Core.

The test bench generates data for the SD-PAL Mode by default. You can generate the core for different input configurations to observe multichannel behavior and FEC correction features of the core.

XILINX

Appendix A

Verification, Compliance, and Interoperability

The SMPTE 2022-5/6 Video over IP Transmitter core has been validated using the Xilinx Kintex®-7 FPGA Connectivity Kit.

See the *SMPTE 2022-5/6 High Bit Rate Media Transport over IP Networks with Forward Error Correction on Kintex-7 FPGAs* (XAPP896) [Ref 6] reference design for more information.

Migrating

This appendix contains information about migrating a design from ISE® to the Vivado® Design Suite, and for upgrading to a more recent version of the IP core. For customers upgrading in the Vivado Design Suite, important details (where applicable) about any port changes and other impact to user logic are included.

Migrating to the Vivado Design Suite

The SMPTE 2022 5/6 Transmitter core version v3.0 has been updated to comply with the latest SMPTE 2022 5/6 Standards Specification. It supports migration from older ISE version (v2.1).

For information about migrating to the Vivado Design Suite, see the *ISE to Vivado Design Suite Migration Guide* (UG911) [Ref 9].

Upgrading in the Vivado Design Suite

No changes have been made to the ports and parameters that affect the core upgrade.

Debugging

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

TIP: *If the IP generation halts with an error, there might be a license issue. See License Checkers in Chapter 1 for more details.*

Finding Help on Xilinx.com

To help in the design and debug process when using the core, 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 core. 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.

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 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 20222-5/6 TX Core

AR 54535

Contacting Technical Support

Xilinx provides technical support at www.xilinx.com/support for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

To contact Xilinx Technical Support:

1. Navigate to www.xilinx.com/support.
2. Open a WebCase by selecting the WebCase link located under Additional Resources.

When opening a WebCase, include:

- Target FPGA including package and speed grade.
- All applicable Xilinx Design Tools and simulator software versions.
- Additional files based on the specific issue might also be required. See the relevant sections in this debug guide for guidelines about which files to include with the WebCase.

Note: Access to WebCase is not available in all cases. Please login to the WebCase tool to see your specific support options.

Vivado Lab Tools

Vivado® lab tools insert logic analyzer and virtual I/O cores directly into your design. Vivado lab tools allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature represents the functionality in the Vivado IDE that is used for logic debugging and validation of a design running in Xilinx FPGA devices in hardware.

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

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

See *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 10].

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. 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 Vivado lab tools capture that the waveform is correct for accessing the AXI4-Lite interface.

AXI4-Stream Interfaces

If data is not being transmitted or received, check the following conditions:

- If transmit `<interface_name>_tready` is stuck Low following the `<interface_name>_tvalid` input being asserted, the core cannot send data.
- If the receive `<interface_name>_tvalid` is stuck Low, the core is not receiving data.
- Check that the `ACLK` inputs are connected and toggling.
- Check that the AXI4-Stream waveforms are being followed.
- Check core configuration.
- Add appropriate core specific checks.

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:

www.xilinx.com/support.

For a glossary of technical terms used in Xilinx documentation, see:

www.xilinx.com/company/terms.htm.

References

These documents provide supplemental material useful with this product guide.

1. *High Bit Rate Media Transport over IP Networks with Forward Error Correction* (XAPP590)
2. *AXI Reference Guide* (UG761)
3. *LogiCORE IP AXI Interconnect Product Guide* (PG059)
4. *LogiCORE IP 10-Gigabit Ethernet MAC Product Guide* (PG072)
5. *Society of Motion Picture and Television Engineers (SMPTE) SD/HD/3G-SDI 2.0 Product Guide* (PG071)
6. *SMPTE2022-5/6 High Bit Rate Media Transport over IP Networks with Forward Error Correction on Kintex-7 FPGAs* (XAPP896)
7. *Vivado Design Suite User Guide: Designing with IP* (UG896)
8. *Vivado Design Suite User Guide: Getting Started* (UG910)
9. *ISE to Vivado Design Suite Migration Guide* (UG911)
10. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
11. *Implementing SMPTE SDI Interfaces with Kintex-7 GTX Transceivers* (XAPP592)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/24/2012	1.0	Initial Xilinx release.
07/25/2012	2.0	Updated to core version 2.0 and added Vivado Design Suite material Added support for Virtex®-7 devices
10/16/2012	2.0.1	Updated memory requirement for core.
12/18/2012	2.1	• Updated to core version 2.1. • Updated to ISE® design tools 14.4 and Vivado Design Suite 2012.4 • Updated design to support the latest SMPTE 2022-5/6 draft change. • Added resource numbers for devices using Vivado Design Suite • Updated screen captures in Chapter 4 and Chapter 6. • Updated Debug appendix.
03/20/2013	3.0	• Revision number advanced to 3.0 to align with core version number. • Updated to core version 3.0 and Vivado Design Suite. • Removed all material related to Virtex-6 devices, ISE Design Suite, CORE Generator tools, and UCF. • Updated Vivado IDE screen capture.
10/02/2013	3.0	• Added XDC and module level constraints to core. • Added demonstration test bench.

Notice of Disclaimer