LogiCORE IP SMPTE2022-5/6 Video over IP Transmitter v2.1

Product Guide

PG032 December 18, 2012

XILINX®

XILINX®

Table of Contents

SECTION II: VIVADO DESIGN SUITE

Chapter 4: Customizing and Generating the Core

Chapter 5: Constraining the Core

SECTION III: ISE DESIGN SUITE

Chapter 6: Customizing and Generating the Core

Chapter 7: Constraining the Core

Chapter 8: Detailed Example Design

SECTION IV: APPENDICES

SECTION I: SUMMARY

Introduction

The Xilinx LogiCORE™ IP SMPTE2022-5/6 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 SMPTE2022-6
- Per stream basis Forward Error Correction (FEC) in accordance to SMPTE2022-5
- 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, Virtex®-7, Kintex™-7, Virtex®-6
Supported User Interfaces	AXI4-Lite, AXI4-Stream, AXI4
Resources	See Table 2-1, Table 2-2, Table 2-3, Table 2-4, Table 2-5, Table 2-6, and Table 2-7
Provided with Core	
Design Files	ISE®: NGC netlist Vivado™: Encrypted HDL
Example Design	XAPP590
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	VHDL or Verilog Structural
Supported S/W Driver	N/A
Tested Design Flows(2)	
Design Entry	ISE Design Suite v14.4 Vivado Design Suite v2012.4(3)
Simulation	Mentor Graphics ModelSim
Synthesis	Xilinx Synthesis Technology (XST) Vivado Synthesis
Support	
Provided by Xilinx @ www.xilinx.com/support	

Notes:

1. For a complete listing of supported devices, see the release notes for this core.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.
3. Supports only 7 series devices.

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 SMPTE2022 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 SMPTE2022-5/6 is aimed at contribution networks (for example, between broadcast center and regional studio).

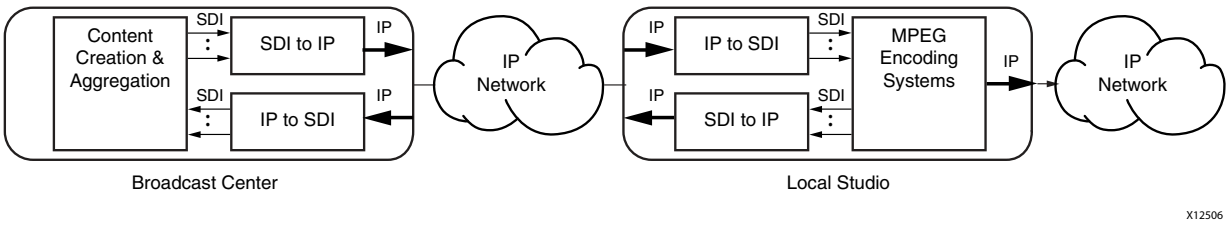

Figure 1-1: **High Bit Rate SMPTE2022-5/6 between Broadcast Center and Local Studio**

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 SMPTE2022-6. For each media stream with SMPTE2022-6, the core creates the Forward Error Correction streams in accordance with SMPTE2022-5 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 CORE Generator™ tool. 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

Operating System Requirements

For operating system requirements, see the Xilinx Design Tools: Release Notes Guide.

Licensing and Ordering Information

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/ISE® 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 SMPTE2022-5/6 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 *AXI Reference Guide* (UG761) for additional information. The function of the core is compliant with *SMPTE 2022-5/6* working draft.

Performance

The following sections detail the performance characteristics of the core.

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, Virtex®-7, Kintex™-7, and Virtex-6 devices in Table 2-1, Table 2-2, Table 2-3, Table 2-4, Table 2-5, Table 2-6, and Table 2-7. These values were generated using Xilinx CORE Generator™ tools, v14.4. They are derived from post-synthesis reports, and might change during MAP and PAR.

ISE Design Suite Resource Utilization Data

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

SDI Channel Number	FEC Include	FFs	LUTs	Slices	LUT FF Pairs	36k BRAMs	18k BRAMs
1	0	8904	6811	2840	10122	16	3
2	0	11919	10153	4130	14253	18	4
3	0	14942	11648	5224	17593	20	5
4	0	17939	13994	6582	21965	22	6
5	0	21044	16818	7828	26451	24	7
6	0	23936	17968	8844	29497	26	8
7	0	26937	19352	10946	35468	28	9
8	0	29980	18872	11536	37878	30	10
1	1	12035	9483	3926	13688	52	8
2	1	15347	13305	5396	18366	54	9
3	1	18744	15463	7456	22124	56	10
4	1	22009	18066	7988	26754	58	11
5	1	25468	21054	9800	32269	60	12
6	1	28699	23006	10054	34666	62	13
7	1	32162	24443	12976	42066	64	14
8	1	35379	24533	13616	44660	66	15

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

SDI Channel Number	FEC Include	FFs	LUTs	Slices	LUT FF Pairs	36k BRAMs	18k BRAMs
1	0	8891	7517	2428	9365	16	3
2	0	11935	10716	4379	14501	18	4
3	0	14887	11728	5106	17589	20	5
4	0	17985	14321	5991	21402	22	6
5	0	21025	16596	8663	27479	24	7
6	0	23936	17871	8750	29554	26	8
7	0	26961	19565	10813	35242	28	9
8	0	30213	19117	11536	38038	30	10
1	1	12022	9471	4261	14308	52	8
2	1	15392	13486	6610	19873	54	9
3	1	18707	15191	8428	22956	56	10
4	1	22024	18105	8229	27223	58	11
5	1	25458	21169	9833	32419	60	12
6	1	28676	22607	11129	36039	62	13
7	1	32140	24806	12641	41609	64	14
8	1	35568	24726	13990	45150	66	15

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

SDI Channel Number	FEC Include	FFs	LUTs	Slices	LUT FF Pairs	36k BRAMs	18k BRAMs
1	0	8891	6937	3035	10488	16	3
2	0	11935	10511	4463	14744	18	4
3	0	14908	11804	4829	17141	20	5
4	0	18007	14063	6405	22047	22	6
5	0	21039	16849	7535	26100	24	7
6	0	23936	18058	8570	29234	26	8
7	0	26961	19409	10959	35640	28	9
8	0	30213	19104	11558	38055	30	10
1	1	12022	9512	4044	13895	52	8
2	1	15392	13744	5533	18583	54	9
3	1	18707	15563	7339	21901	56	10
4	1	22024	18112	7818	26909	58	11
5	1	25458	21266	9617	32118	60	12
6	1	28676	22822	10468	35301	62	13
7	1	32140	24758	12419	41235	64	14
8	1	35568	24706	13421	44570	66	15

Table 2-4: **Resource Utilization for Virtex-6 Families (6vcx130t Speed -1)**

SDI Channel Number	FEC Include	FFs	LUTs	Slices	LUT-FF pairs	36k BRAMs	18k BRAMs
1	0	8909	7221	3245	10001	16	3
2	0	11889	10302	5224	14889	18	4
3	0	14912	12236	4900	16023	20	5
4	0	17937	13961	7337	21631	22	6
5	0	20934	17354	7190	24400	24	7
6	0	24001	18187	9691	29232	26	8
7	0	27062	19609	11919	35217	28	9
8	0	30153	19703	12317	36905	30	10
1	1	12000	9674	5079	14174	52	8
2	1	15337	13493	6890	19158	54	9
3	1	18704	15457	7656	22083	56	10
4	1	22020	18064	9079	26500	58	11
5	1	25427	21045	11279	32080	60	12
6	1	28676	23923	9499	32181	62	13
7	1	32118	24705	13797	41018	64	14
8	1	35495	24594	14754	43974	66	15

Vivado Design Suite Resource Utilization Data

Table 2-5: **Resource Utilization for Zynq-7000 Devices (xc7z045 Speed -1)**

SDI Channel Number	FEC Include	FFs	LUTs	Slices	LUT-FF pairs	36k BRAMs	18k BRAMs
1	0	8970	8762	3099	10241	16	2
2	0	11977	11106	4384	13752	18	2
3	0	15002	12765	5550	16924	20	2
4	0	18060	14283	6204	19599	22	2
5	0	21061	17306	8043	24094	24	2
6	0	24090	18744	9063	27743	26	2
7	0	27145	21838	10363	30522	28	2
8	0	30167	21072	11192	33548	30	2
1	1	11613	11569	4185	13669	52	7
2	1	15009	14392	5774	17764	54	7
3	1	18373	16407	6834	21210	56	7
4	1	21694	18318	8282	25013	58	7
5	1	25099	21716	9517	29094	60	7
6	1	28416	23554	10838	33499	62	7
7	1	31793	27009	12145	36477	64	7
8	1	35137	26609	12543	39204	66	7

Table 2-6: **Resource Utilization for Virtex-7 FPGAs (xc7vx690t Speed -1)**

SDI Channel Number	FEC Include	FFs	LUTs	Slices	LUT-FF pairs	36k BRAMs	18k BRAMs
1	0	8948	8734	3188	10482	16	2
2	0	12002	11125	4325	13760	18	2
3	0	15002	12755	5118	16488	20	2
4	0	18029	14265	6265	19722	22	2
5	0	21061	17311	7496	23465	24	2
6	0	24127	18766	9042	27561	26	2
7	0	27145	21820	10211	30323	28	2
8	0	30167	21078	10660	32948	30	2
1	1	11613	11571	4304	13852	52	7
2	1	14984	14365	5849	17840	54	7
3	1	18345	16396	6741	21132	56	7
4	1	21694	18322	8330	24867	58	7
5	1	25099	21723	9187	28748	60	7
6	1	28453	23582	10788	33218	62	7
7	1	31771	26985	11735	36049	64	7
8	1	35137	26622	12815	39643	66	7

Table 2-7: **Resource Utilization for Kintex-7 FPGAs (xc7k325t Speed -1)**

SDI Channel Number	FEC Include	FFs	LUTs	Slices	LUT-FF pairs	36k BRAMs	18k BRAMs
1	0	8948	8743	3110	10443	16	2
2	0	11977	11108	4482	13858	18	2
3	0	15030	12783	5749	17182	20	2
4	0	18060	14290	6331	19872	22	2
5	0	21095	17333	7579	23654	24	2
6	0	24127	18766	8938	27405	26	2
7	0	27145	21830	9751	30150	28	2
8	0	30145	21055	11188	33577	30	2
1	1	11635	11596	4463	14175	52	7
2	1	15009	14394	5937	17842	54	7
3	1	18345	16393	6980	21368	56	7
4	1	21694	18319	8081	24823	58	7
5	1	25099	21712	9256	28918	60	7
6	1	28453	23587	10718	33211	62	7
7	1	31793	27012	11942	36464	64	7
8	1	35115	26599	12645	39145	66	7

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 GUI.

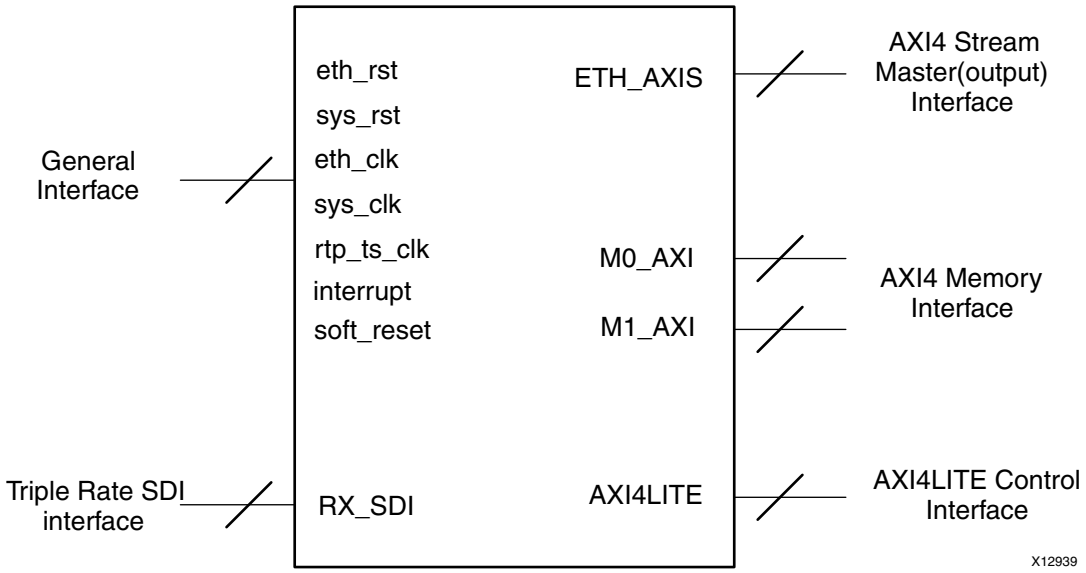

Figure 2-1: **Core Top Level Signaling Interface**

General Interface

Table 2-8 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 2-8: **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) for more information.

Table 2-9: **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 2-9: **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) for more information.

Table 2-10: **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.

Triple-Rate SDI Interface

See the *LogiCORE IP Virtex-6 FPGA Triple-Rate SDI User Guide* (UG823) for more information.

Table 2-11: **Triple Rate SDI Interface Signals**

Signal Name	Direction	Width	Description
rx_rst	In	1	Reset
rx_clk	In	1	Connect to rx_usrclk of Triple-Rate SDI core.
rx_mode_locked	In	1	Connect to rx_mode_locked of Triple-Rate SDI core.
rx_locked	In	1	Connect to rx_t_locked of Triple-Rate SDI core.
rx_t_family	In	4	Connect to rx_t_family of Triple-Rate SDI core.
rx_t_rate	In	4	Connect to rx_t_tate of Triple-Rate SDI core.
rx_bit_rate	In	1	Connect to rx_bit_rate of Triple-Rate SDI core.
rx_mode	In	2	Connect to rx_mode of Triple-Rate SDI core.
rx_eav	In	1	Connect to rx_eav of Triple-Rate SDI core.
rx_ce_sd	In	1	Connect to rx_ce_sd of Triple-Rate SDI core.
rx_dout_rdy_3g	In	1	Connect to rx_dout_rdy_3g of Triple-Rate SDI core.
rx_crc_err_a	In	1	Connect to rx_crc_err_a of Triple-Rate SDI core.
rx_a_vpid_valid	In	1	Connect to rx_a_vpid_valid of Triple-Rate SDI core.
rx_a_vpid	In	32	Connect to rx_a_vpid of Triple-Rate SDI core.
rx_line_a	In	11	Connect to rx_line_a of Triple-Rate SDI core.

Table 2-11: **Triple Rate SDI Interface Signals *(Cont'd)***

Signal Name	Direction	Width	Description
rx_ds1_a	In	10	Connect to rx_ds1a of Triple-Rate SDI core.
rx_ds2_a	In	10	Connect to rx_ds2a of Triple-Rate SDI core.
rx_ds1_b	In	10	Connect to rx_ds1b of Triple-Rate SDI core.
rx_ds2_b	In	10	Connect to rx_ds2b of Triple-Rate SDI core.
rx_level_b_3g	In	1	Connect to rx_level_b_3g of Triple-Rate 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 SMPTE2022-5/6 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 2-12: **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.
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.

Table 2-12: **AXI4-Lite Interface Signals *(Cont'd)***

Signal Name	Direction	Width	Description
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 2-13: **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	0x02010000	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 2-13: **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 triple rate 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 Triple-Rate 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 SMPTE2022-5/6, 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 *XAPP590* *High Bit Rate Media Transport over IP Networks with Forward Error Correction* reference design for more information.

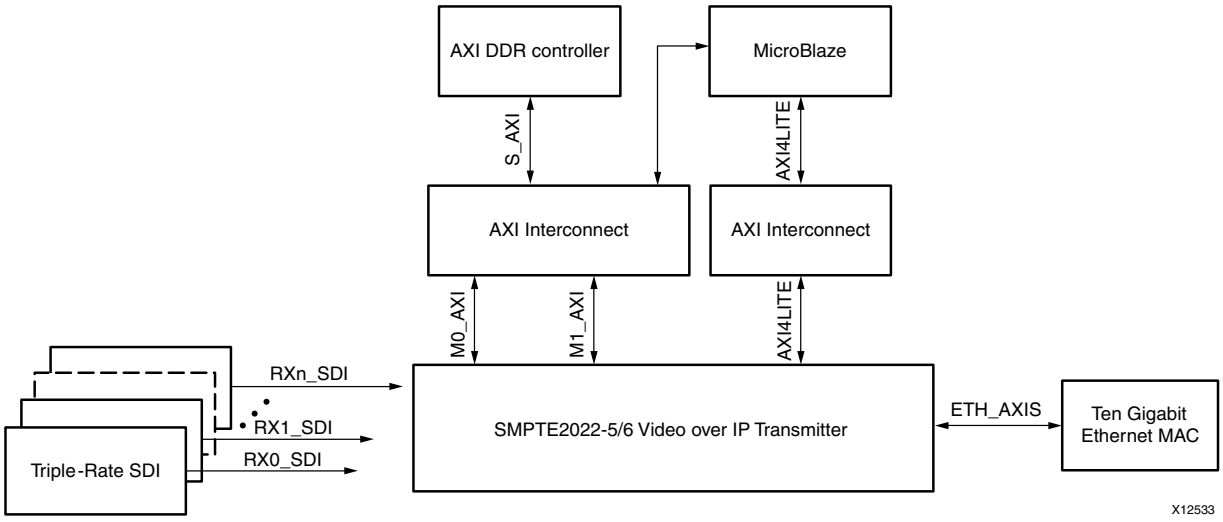

Figure 3-1: **SMPTE2022-5/6 Video over IP Transmitter System Built with other Xilinx IP Cores**

Note: There is an option to include Forward Error Correction engine in the SMPTE2022-5/6 Video over IP Transmitter core. Adding this will enable 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 three clock domains.

- SDI video clock domain
- System clock domain recommended running at 200 MHz
- Ethernet clock domain at 156.25 MHz for 10 Gb/s bandwidth

Resets

See the *XAPP590* *High Bit Rate Media Transport over IP Networks with Forward Error Correction* reference design for more information.

Memory Requirement

Table 3-1 shows a tabulation of the amount of DDR memory required by the SMPTE2022-5/6 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 SMPTE2022-5/6 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

SECTION II: VIVADO DESIGN SUITE

Customizing and Generating the Core

Constraining the Core

Customizing and Generating the Core

This chapter includes information about using Xilinx tools to customize and generate the core in the Vivado™ Design Suite. For more information about the Vivado Design Suite, see the Vivado Design Suite - 2012.4 User Guides web page.

GUI

The core is configured to meet the developer's specific needs before instantiation through the Vivado integrated design environment (IDE). This section provides a quick reference to parameters that can be configured at generation time.

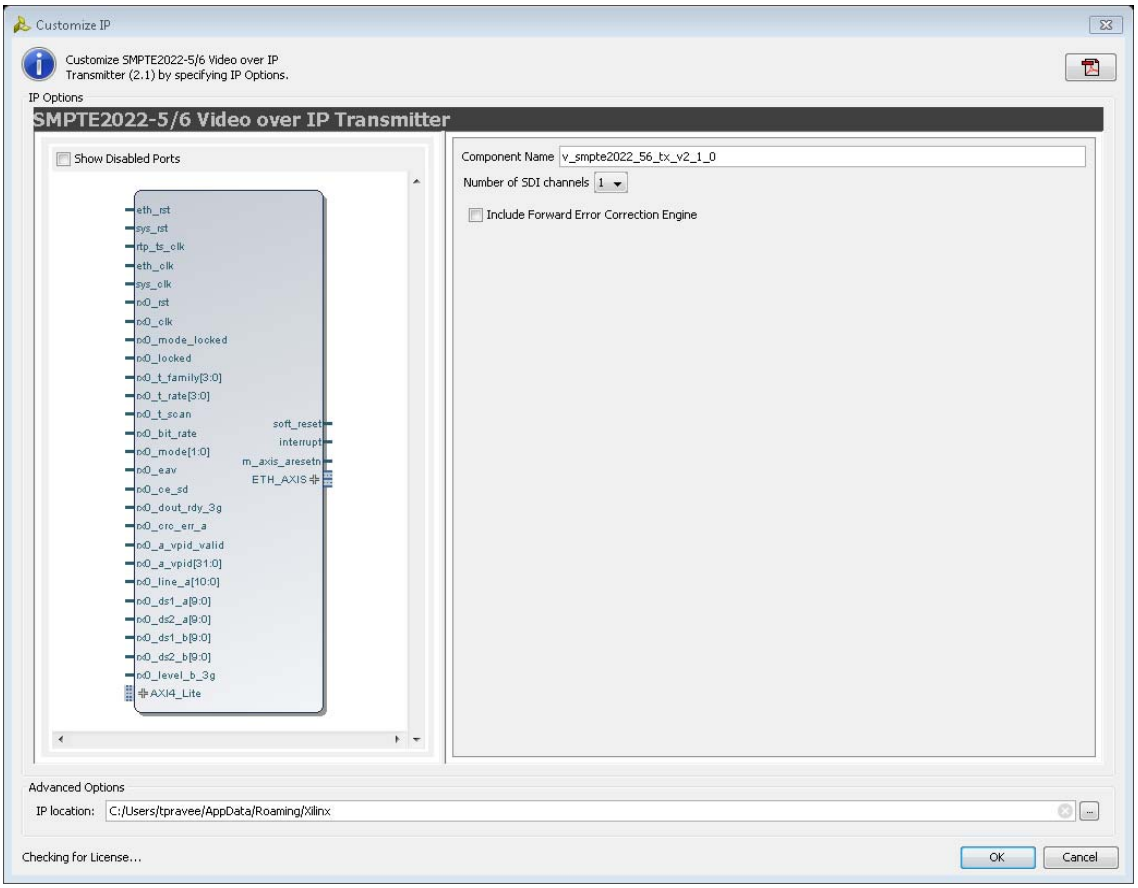

Figure 4-1: **Vivado IDE**

The GUI displays 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.

The Vivado design tools output consists of some or all the following files.

Table 4-1: **File Details**

Name	Description
<component_name>	Library directory for the v_smpte2022_56_tx core which contains the encrypted source files.
<blk_mem_gen_v7_3>	Library directory for the helper core which contains the encrypted source files.
<component_name>.vho <component_name>.veo	The HDL template for instantiating the core.
<component_name>.xci	IP-XACT file describing which options were used to generate the core. An XCI file can also be used as a source file for designs created with the Vivado Design Suite.
<component_name>.xml	IP-XACT XML file describing how the core is constructed so Vivado design tools can properly build the core.

Constraining the Core

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

Required Constraints

There are no required constraints for this 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

The core has three clock domains.

- SDI clock domain
- System clock domain recommended running at 200 MHz
- Ethernet clock domain at 156.25 MHz.

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.

SECTION III: ISE DESIGN SUITE

Customizing and Generating the Core

Constraining the Core

Detailed Example Design

Customizing and Generating the Core

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

GUI

The core is configured to meet the developer's specific needs through the CORE Generator™ Graphical User Interface (GUI). This section provides a quick reference to parameters that can be configured at generation time.

Figure 6-1: **CORE Generator Tool Graphical User Interface**

The GUI displays 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 **Yes** is selected the core is generated with Forward Error Correction engine.

Parameter Values in the XCO File

Table 6-1 defines valid entries for the Xilinx CORE Generator tool (XCO) parameters. Xilinx strongly suggests that XCO parameters are not manually edited in the XCO file; instead, use the CORE Generator system GUI to configure the core and perform range and parameter value checking. The XCO parameters are helpful in defining the interface to other Xilinx tools.

Table 6-1: **XCO Parameters**

XCO Parameter	Default	Valid Values
component_name	smpte2022_56_voip_tx	ASCII text using characters: a..z, 0..9 and "_" starting with a letter. ***Note:*** "v_smpte2022_56_tx" is not allowed.
c_sdi_channels	1	1-8
c_include_fec	0	0,1
c_phy	0	0
c_sim_mode	0	0
c_debug_mode	0	0

Output Generation

The Xilinx CORE Generator tool for the SMPTE2022-5/6 Video over IP Transmitter core output the core as a netlist that can be instantiated directly in a HDL design. The output is placed in the <project directory>. The CORE Generator tool output consists of some or all of the following files:

Table 6-2: **File Details**

Name	Description
<component_name>_readme.txt	Readme file for the core
<component_name>.ngc	The netlist for the core
<component_name>.vho	The HDL template for instantiating the core
<component_name>.vhd	The structural simulation model for the core. It is used for functionally simulating the core.
<component_name>.xco	Log file from CORE Generator system describing which options were used to generate the core. An XCO file can also be used as an input to the CORE Generator tool.
<component_name>_flist.txt	A text file listing all of the output files produced when the customized core was generated in the CORE Generator tool.
<component_name>.asy	IP symbol file.
<component_name>.gise <component_name>.xise	ISE® design tools subproject files for use when including the core in ISE designs.

Constraining the Core

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

Required Constraints

There are no required constraints for this 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

The core has three clock domains.

- SDI clock domain
- System clock domain recommended running at 200 MHz
- Ethernet clock domain at 156.25 MHz.

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.

Detailed Example Design

No example design is available for the v2.1 core. See the XAPP590 *High Bit Rate Media Transport over IP Networks with Forward Error Correction* reference design for more information.

SECTION IV: APPENDICES

XILINX®

Appendix A

Verification, Compliance, and Interoperability

Hardware Testing

The SMPTE2022-5/6 Video over IP Transmitter core has been validated using Xilinx Virtex-6 FPGA Broadcast Connectivity Kit. See the XAPP590 *High Bit Rate Media Transport over IP Networks with Forward Error Correction* reference design for more information.

Appendix B

Migrating

See the *Vivado Design Suite Migration Methodology Guide*. (UG911). For more information about the Vivado Design Suite, see the Vivado Design Suite - 2012.4 User Guides web page.

Debugging

This appendix includes details about resources available on the Xilinx Support website and debugging tools. In addition, this appendix provides a step-by-step debugging process and a flow diagram to guide you through debugging the core.

The following topics are included in this appendix:

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 opening a Technical Support WebCase.

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.

Release Notes

Known issues for all cores, including this core are described in the IP Release Notes Guide (XTP025).

Known Issues

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.

Answer Record for the SMPTE20222-5/6 TX Core

See 47207. This web page provides links to Answer Records associated with this core.

Contacting Technical Support

Xilinx provides premier technical support for customers encountering issues that require additional assistance.

To contact Xilinx Technical Support:

1. Navigate to www.xilinx.com/support.
2. Open a WebCase by selecting the WebCase link located under Support Quick Links.

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.

License Checkers

If the IP requires a license key, the key must be verified. The ISE® and 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 tools:

- ISE flow: XST, NgdBuild, Bitgen
- Vivado flow: Vivado Synthesis, Vivado Implementation, write_bitstream (Tcl 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 time out.

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 ChipScope™ debugging tool 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.

Appendix D

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. Unless otherwise noted, IP references are for the product documentation page.

1. Vivado™ Design Suite user documentation (www.xilinx.com/cgi-bin/docs/rdoc?v=2012.4;t=vivado+docs)
2. *AXI Reference Guide* (UG761)
3. *LogiCORE IP 10-Gigabit Ethernet MAC Product Guide* (PG072)
4. *LogiCORE IP Virtex-6 FPGA Triple-Rate SDI User Guide* (UG823)
5. *Vivado Design Suite Migration Methodology Guide* (UG911)
6. *LogiCORE IP AXI Interconnect Product Guide* (PG059)
7. *High Bit Rate Media Transport over IP Networks with Forward Error Correction* (XAPP590)
8. *Implementing SMPTE SDI Interfaces with Kintex-7 GTX Transceivers* (XAPP592)

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.

See the *IP Release Notes Guide* (XTP025) for more information on this core. For each core, there is a master Answer Record that contains the Release Notes and Known Issues list for the core being used. The following information is listed for each version of the core:

- New Features
- Resolved Issues
- Known Issues

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/24/12	1.0	Initial Xilinx release.
07/25/12	2.0	Updated to core version 2.0 and added Vivado™ Design Suite material Added support for Virtex®-7 devices
10/16/12	2.0.1	Updated memory requirement for core.
12/18/12	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.

Notice of Disclaimer