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

Product Guide for Vivado Design Suite

PG033 October 2, 2013

XILINX®

XILINX

Table of Contents

Introduction

The Xilinx LogiCORE™ IP SMPTE 2022-5/6 Video over IP Receiver is a module for broadcast applications that requires bridging between SMPTE video connectivity standards bridging between uncompressed SMPTE video connectivity standards (SD/HD/3G-SDI) and 10Gb/s IP networks. The module is capable of recovering IP packets lost due to network transmission errors and ensure the picture quality of uncompressed, high bandwidth professional video is maintained. The core is for developing Internet protocol-based systems to reduce overall cost in broadcast facilities for distribution and routing of audio and video data.

Features

- Handle up to 8 channels of SD/HD/3G-SDI streams (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
- Supports Virtual Local Area Network (VLAN)
- AXI4-Stream data interfaces
- AXI4-Lite control interface
- Configurable channel selection based on IP source address, User Datagram Protocol (UDP) destination port, and Real-time Transport Protocol (RTP) Synchronization Source (SSRC) identifier over AXI4-Lite interface
- Supports SD-SDI, HD-SDI, 3G-SDI Level-A, 3G-SDI Level-B single stream and 3G-SDI Level-B dual stream

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	Zynq®-7000, 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	SMPTE 2022-*5/6 High Bit Rate Media Transport over IP Networks with Forward Error Correction on Kintex-7 FPGAs* (XAPP896) [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 telecommunication 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 ensuring 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).

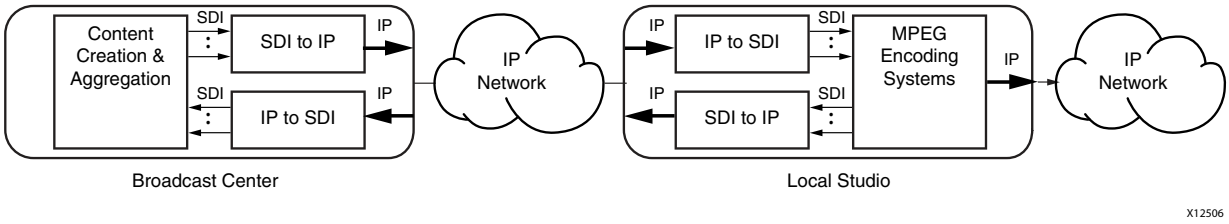

Figure 1-1: **High Bit Rate SMPTE 2022-5/6 between Broadcast Center and Local Studio**

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 help 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 enable remote production from existing fixed studio set ups, which dramatically reduces both capital expenditure and operating expenses.

Feature Summary

The core maps Ethernet packets into raw SD/HD/3G-SDI video streams and is capable of recovering IP packets lost to network transmission errors to 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 tools. 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 license check succeeds, the IP can continue generation. Otherwise, generation halts with error. License checkpoints are enforced by the following 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 video 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.

IMPORTANT: *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 Design 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 the this core have been estimated for the devices shown in Table 2-1, Table 2-2, and Table 2-3. These values were generated using the 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	7,698	6,549	2,745	8,456	14	2	0
2	0	10,585	8,676	3,624	11,351	21	3	0
3	0	13,434	10,104	4,836	14,329	28	4	0
4	0	16,305	10,717	5,502	16,571	35	5	0
5	0	19,145	12,132	6,194	19,046	42	6	0
6	0	22,016	13,429	7,564	21,972	49	7	0
7	0	24,876	14,480	8,786	25,012	56	8	0
8	0	27,743	15,253	9,827	28,044	63	9	0
1	1	12,900	9,238	3,959	12,649	50	7	0
2	1	16,398	11,503	5,239	16,011	57	9	0
3	1	20,042	13,654	6,607	19,601	78	12	0
4	1	23,656	14,913	7,770	22,798	85	15	0
5	1	27,284	16,973	8,821	26,194	120	21	0
6	1	30,893	19,140	10,677	30,144	127	21	0
7	1	34,491	20,447	10,434	32,144	134	24	0
8	1	38,105	21,569	12,152	36,419	141	27	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	7,698	6,546	2,531	8,380	14	2	0
2	0	10,585	8,665	3,643	11,341	21	3	0
3	0	13,434	10,103	4,756	14,258	28	4	0
4	0	16,305	10,702	5,637	16,702	35	5	0
5	0	19,145	12,140	6,501	19,299	42	6	0
6	0	22,016	13,418	7,516	21,975	49	7	0
7	0	24,876	14,492	9,012	25,161	56	8	0
8	0	27,743	15,233	8,710	26,901	63	9	0
1	1	12,900	9,248	4,109	12,826	50	7	0
2	1	16,398	11,506	5,364	16,000	57	9	0
3	1	20,042	13,649	6,973	19,947	78	12	0
4	1	23,656	14,892	8,093	23,159	85	15	0
5	1	27,284	16,976	10,833	27,784	120	21	0
6	1	30,893	19,142	11,032	30,589	127	21	0
7	1	34,491	20,451	13,137	34,748	134	24	0
8	1	38,105	21,571	13,310	37,788	141	27	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	7,698	6,542	2,780	8,551	14	2	0
2	0	10,585	8,675	3,725	11,479	21	3	0
3	0	13,434	10,103	4,767	14,278	28	4	0
4	0	16,305	10,714	5,415	16,390	35	5	0
5	0	19,145	12,133	6,556	19,221	42	6	0
6	0	22,016	13,420	6,951	21,446	49	7	0
7	0	24,876	14,476	7,967	24,300	56	8	0
8	0	27,743	15,249	9,890	28,081	63	9	0
1	1	12,900	9,241	3,959	12,643	50	7	0
2	1	16,380	11,476	5,440	16,129	57	9	0
3	1	20,042	13,639	6,830	19,740	78	12	0
4	1	23,656	14,905	8,023	23,092	85	15	0
5	1	27,284	16,967	8,338	25,812	120	21	0
6	1	30,893	19,134	11,145	30,509	127	21	0
7	1	34,491	20,446	11,446	33,424	134	24	0
8	1	38,105	21,577	12,322	36,598	141	27	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 2-2 shows an I/O Diagram of the core. The SDI_TX interface pins depend on the number of channels configured through the Vivado Integrated Design Environment (IDE).

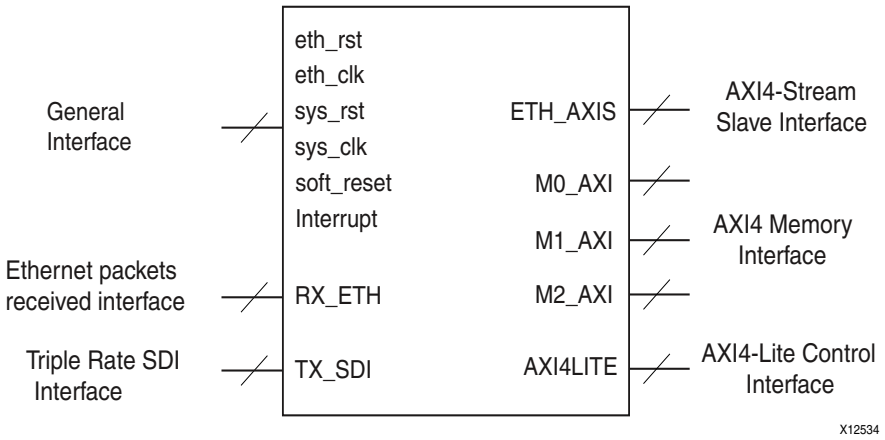

Figure 2-2: **SMPTE 2022-5/6 Video over IP Receiver Core Top Level Signaling Interface**

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: **General Interface Signals**

Signal Name	Direction	Width	Description
eth_rst	In	1	Ethernet domain reset.
eth_clk	In	1	156.25 MHz Ethernet clock.
sys_rst	In	1	System domain reset.
sys_clk	In	1	200MHz system clock.
interrupt	Out	1	Reserved
soft_reset	Out	1	Core reset from the control register

AXI4 Memory Interface

The core uses an AXI4 interface to connect to the AXI4 interconnect. The AXI4 Interconnect provides the access to the external memory through the 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_awprot	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_axi_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

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

Signal Name	Direction	Width	Description
m0_axi_arlock	Out	2	Read Address Channel Atomic Access Type
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_awid	Out	1	Write Address Channel Transaction ID
m1_axi_awaddr	Out	32	Write Address Channel Address
m1_axi_awlen	Out	8	Write Address Channel Burst Length code
m1_axi_awsize	Out	3	Write Address Channel Transfer Size code
m1_axi_awburst	Out	2	Write Address Channel Burst Type
m1_axi_awlock	Out	2	Write Address Channel Atomic Access Type
m1_axi_awcache	Out	4	Write Address Channel Cache Characteristics
m1_axi_awprot	Out	3	Write Address Channel Protection Bits
m1_axi_awqos	Out	4	Write Address Channel Quality of Service
m1_axi_awvalid	Out	1	Write Address Channel Valid
m1_axi_awready	In	1	Write Address Channel Ready
m1_axi_wdata	Out	256	Write Data Channel Data
m1_axi_wstrb	Out	32	Write Data Channel Data Byte Strobes
m1_axi_wlast	Out	1	Write Data Channel Last Data Beat
m1_axi_wvalid	Out	1	Write Data Channel Valid
m1_axi_wready	In	1	Write Data Channel Ready
m1_axi_bid	In	1	Write Response Channel Transaction ID

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

Signal Name	Direction	Width	Description
m1_axi_bresp	In	2	Write Response Channel Response Code
m1_axi_bvalid	In	1	Write Response Channel Valid
m1_axis_bready	Out	1	Write Response 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	Out	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
m2_axi_awid	Out	1	Write Address Channel Transaction ID
m2_axi_awaddr	Out	32	Write Address Channel Address
m2_axi_awlen	Out	8	Write Address Channel Burst Length code
m2_axi_awsize	Out	3	Write Address Channel Transfer Size code
m2_axi_awburst	Out	2	Write Address Channel Burst Type
m2_axi_awlock	Out	2	Write Address Channel Atomic Access Type
m2_axi_awcache	Out	4	Write Address Channel Cache Characteristics
m2_axi_awprot	Out	3	Write Address Channel Protection Bits
m2_axi_awqos	Out	4	Write Address Channel Quality of Service

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

Signal Name	Direction	Width	Description
m2_axi_awvalid	Out	1	Write Address Channel Valid
m2_axi_awready	In	1	Write Address Channel Ready
m2_axi_wdata	Out	256	Write Data Channel Data.
m2_axi_wstrb	Out	32	Write Data Channel Data Byte Strobes
m2_axi_wlast	Out	1	Write Data Channel Last Data Beat
m2_axi_wvalid	Out	1	Write Data Channel Valid
m2_axi_wready	In	1	Write Data Channel Ready
m2_axi_bid	In	1	Write Response Channel Transaction ID
m2_axi_bresp	In	2	Write Response Channel Response Code
m2_axi_bvalid	In	1	Write Response Channel Valid
m2_axi_bready	Out	1	Write Response Channel Ready

AXI4-Stream Slave Interface

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
s_axis_aresetn	Out	1	AXI4-Stream active-Low reset for Receive path - 10 Gigabit Ethernet MAC (XGMAC)
s_axis_tdata[63:0]	In	64	AXI4-Stream Data from XGMAC
s_axis_tkeep[7:0]	In	8	AXI4-Stream Data Control from XGMAC
s_axis_tvalid	In	1	AXI4-Stream Data Valid from XGMAC
s_axis_tlast	In	1	AXI4-Stream signal from XGMAC indicating an end of packet
s_axis_tuser	In	1	AXI4-Stream User Sideband Interface from XGMAC • 1 indicates that a good packet has been received. • 0 indicates that a bad packet has been received.

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
tx_rst	In	1	Reset.
tx_clk	In	1	Clock input. It must have a frequency of 74.25 MHz or 74.25/1.001 MHz for HD-SDI, 148.5 MHz or 148.5/1.001 MHz for 3G-SDI, and 148.5 MHz for SD-SDI mode.
tx_ce	Out	3	To tx_ce of SMPTE SD/HD/3G-SDI
tx_din_rdy	Out	1	To tx_din_rdy of SMPTE SD/HD/3G-SDI
tx_ds1a	Out	10	To tx_ds1a of SMPTE SD/HD/3G-SDI
tx_ds1b	Out	10	To tx_ds1b of SMPTE SD/HD/3G-SDI
tx_ds2a	Out	10	To tx_ds2a of SMPTE SD/HD/3G-SDI
tx_ds2b	Out	10	To tx_ds2b of SMPTE SD/HD/3G-SDI
tx_level_b_3g	Out	1	To tx_level_b_3g of SMPTE SD/HD/3G-SDI
tx_mode	Out	1	To tx_mode of SMPTE SD/HD/3G-SDI
tx_m	Out	1	In HD-SDI and 3G-SDI modes, this output indicates which bit rate is received. If this output is Low, it indicates a bit rate of 1.485 Gb/s in HD-SDI mode and 2.97 Gb/s in 3G-SDI mode. If this output is High, it indicates a bit rate of 1.485/1.001 Gb/s in HD-SDI mode and 2.97/1.001 Gb/s in 3G-SDI mode.

Ethernet Packets Received Interface

See the *SMPTE 2022-5/6* reference design for more information.

Table 2-8: **Ethernet Packets Received Interface Signals**

Signal Name	Direction	Width	Description
rx_rtp_pkt_recv	Out	1	Pulse indicating receiving of RTP packets
rx_rtp_seq_num	Out	16	Sequence number of RTP packet received
rx_rtp_pkt_buffered	Out	16	Amount of RTP packets buffered
rx_rtp_pkt_transmit	Out	1	Pulse indicating consumption of RTP packet for SDI output
rx_vid_lock	Out	1	Indication of channel locking to certain video payload
rx_rtp_vid_ts	Out	32	Video timestamp of the RTP packet received.
rx_rtp_ts	Out	32	RTP timestamp of the RTP packet received

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 Receiver 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. See the *LogiCORE IP AXI Interconnect Product Guide* (PG059) [Ref 3] for more information.

Table 2-9: **AXI4-Lite Interface Signals**

Signal Name	Direction	Width	Description
s_axi_aclk	In	1	AXI4-Lite 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_bready	In	1	AXI4-Lite Write Response Channel Ready. Indicates target is ready to receive response.
s_axi_araddr	In	9	AXI4-Lite Read Address Bus
s_axi_arvalid	In	1	AXI4-Lite Read Address Channel Read Address Valid
s_axi_rready	In	1	AXI4-Lite Read Data Channel Read Data Ready. Indicates target is ready to accept the read data.
s_axi_arready	Out	1	AXI4-Lite Read Address Channel Read Address Ready. Indicates target is ready to accept the read address.
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
s_axi_wready	Out	1	AXI4-Lite Write Data Channel Write Data Ready. Indicates target 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.
s_axi_awready	Out	1	AXI4-Lite Write Address Channel Write Address Ready.

Register Space

The SMPTE 2022-5/6 Video over IP Receiver 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 2-10: **AXI4-Lite Register Map**

Address (hex)	Register Name	Access Type	Default Value	Register Description
General Registers				
0x000	CONTROL	R/W	0	Bit 0: Reserved Bit 1: Register update Bit 31-2: Reserved
0x004	RESET	R/W	0	Bit 0: Soft reset Bit 31-1: Reserved
0x030	CHANNEL	R/W	0	Bit 31-0: Access channel
0x03C	VERSION	R	0x03000000	Bit 7-0: Revision number Bit 11-8: Patch ID Bit 15-12: Version revision Bit 23-16: Version minor Bit 31-24: Version major
0x050	AXI_MM_ADDR_MSB	R/W	0	Bit 2-0: Most significant three bits of the 32-bit AXI memory map address to access the DDR through the AXI interconnect Bit 31-3: Reserved
0x054	DELAY	R/W	0x000186A0	Bit 31:0: Reserved
0x0A0	NUM_CHAN	R	0	Bit 10-0: Number of channels Bit 31-11: Reserved

Table 2-10: **AXI4-Lite Register Map *(Cont'd)***

Address (hex)	Register Name	Access Type	Default Value	Register Description
Channel Registers				
0x100	CHAN_EN	R/W	0	Bit 0: Channel Enable Bit 31-1: Reserved
0x110	FIREWALL_SEL	R/W	0	Bit 1- 0: Select which firewall parameters to be used to filter the Ethernet packets. "00"- dest_port, "01"- dest_port and src_ip, "10"- dest_port and ssrc, "11"- dest_port, src_ip and ssrc" Bit 31-2: Reserved
0x114	DEST_PORT	R/W	0	Bit 15-0: UDP destination port Bit 31-16: Reserved
0x118	SSRC	R/W	0	Bit 31-0: Synchronization Source (SSRC) value
0x11C	SRC_IP_HOST_ADDR	R/W	0	Bit 31-0: Source IP address
0x12C	START_BUFFER_SIZE	R/W	0	Bit 31-0: The number of RTP packets to buffer before SDI playout
0x144	VID_SRC_FMT	R	0	Bit 31-0: Video source format value
0x148	VID_LOCK_PARAM	R	0	Bit 0: Video locked Bit 31-1: Reserved
0x154	FEC_L	R	0	Bit 9-0: L value Bit 31-10: Reserved
0x158	FEC_D	R	0	Bit 9-0: D value Bit 31-10: Reserved
0x15C	FEC_LOCK_PARAM	R	0	Bit 0: FEC locked Bit 1: FEC protect level. '0' - Level A, '1' - Level B. Bit 31-2: Reserved
0x160	PACKETS_BUFFERED	R	0	Bit 15-0: Number of RTP packets buffered
0x180	SDI_STATUS	R	0	Bit 0: Frame error Bit 31-1: Reserved

CONTROL (0x000) 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 (0x004) Register

Bit 0 is software reset. When High, all the other registers and the core are held at reset state.

CHANNEL (0x030) Register

Set the channel registers to access. All the channels share the same set of register address in the channel space.

VERSION (0x03C) Register

Bit fields of the 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 (0x050) Register

The core will attempt to use up to 512 MB of memory leaving the most significant three bits of the 32-bit AXI memory map address for user control.

NUM_CHAN (0x0A0) Register

This register indicates the number of channels in the design.

CHAN_EN (0x100) Register

Set High to turn on the channel operation.

FIREWALL_SEL (0x110) Register

Configures the channel to filter the Ethernet packets based on DEST_PORT, SSRC or SRC_IP_HOST_ADDR registers.

DEST_PORT (0x114) Register

Configures the UDP destination port, a parameter that is used to filter the Ethernet packets for the channel.

SSRC (0x118) Register

Configures the Synchronization Source identifier, a parameter that is used to filter the Ethernet packets for the channel.

SRC_IP_HOST_ADDR (0x11C) Register

Configures the source IP address, a parameter that is used to filter the Ethernet packets for the channel.

START_BUFFER_SIZE (0x12C) Register

Configures the latency of the output SDI based on the number of RTP packets to accumulate before starting.

VID_SRC_FMT (0x144) Register

This register contains the video payload identifier of the SDI video format received based on SMPTE 352M standard. It is valid when the video locked bit is High.

VID_LOCKED_PARAM (0x148) Register

Channel is locked to certain video source payload when video locked bit is High.

FEC_L (0x154) Register

This register contains the L value of FEC matrix. It is valid when FEC locked bit is High.

FEC_D (0x158) Register

This register contains the D value of FEC matrix. It is valid when FEC locked bit is High.

FEC_LOCKED_PARAM (0x15C) Register

FEC locked bit High indicates receiver has received FEC packets with certain L and D configuration.

FEC protection level bit indicates if the channel uses one FEC stream (Level A) or two FEC stream (Level B).

PACKETS_BUFFERED (0x160) Register

Read back on the number of packets currently being buffered in the external DDR memory.

SDI_STATUS (0x180) Register

Received incorrect amount of packets per frame when frame error bit is High. Reset the core to ensure proper operation.

Designing with the Core

The core is for broadcast applications that require bridging between SMPTE video connectivity standards SD/HD/3G-SDI and 10Gb/s Ethernet. The core takes in Ethernet packets encapsulated in accordance with SMPTE 2022-5/6 and maps them in uncompressed SD/HD/3G-SDI streams to the SMPTE SD/HD/3G-SDI core. It receives Ethernet packets through the AXI4-Stream interface from the 10 Gb/s Ethernet MAC. The core uses the AXI4 memory interface to transfer data between the core and external DDR memory. The register control interface is compliant with AXI4-Lite interface. See *SMPTE 2022-5/6 High Bit Rate Media Transport over IP Networks with Forward Error Correction on Kintex-7 FPGAs* (XAPP896) [Ref 1] for more information.

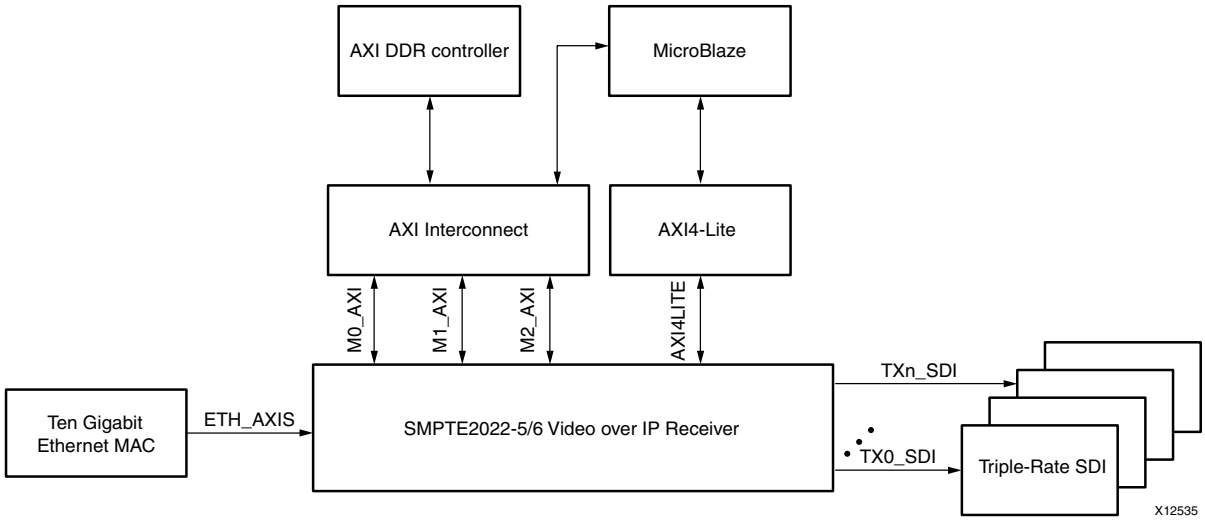

Figure 3-1: **SMPTE 2022-5/6 Video over IP Receiver System Built with other Xilinx IP Cores**

Note: There is an option to include Forward Error Correction engine in the SMPTE 2022-5/6 Video over IP Receiver 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 for 10Gb/s bandwidth.
- AXI4-Lite clock domain recommended at 100 MHz

Resets

See the *SMPTE 2022-5/6 High Bit Rate Media Transport over IP Networks with Forward Error Correction on Kintex-7 FPGAs* (XAPP896) [Ref 1].

Memory Requirement

Table 3-1 shows tabulation of the amount of DDR memory required by the SMPTE 2022-5/6 Video over IP Receiver core based on the number of channels instantiated in the design.

Table 3-1: **Memory Requirement for the SMPTE 2022-5/6 Video over IP Receiver Core**

Number of Channels Instantiated	Size of DDR Memory Needed (MB)
1	128
2	256
3	384
4	512
5	320
6	384
7	448
8	512

Customizing and Generating the Core

This chapter includes information on using Xilinx tools to customize and generate the core using 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 tool bar 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 6] and the "Working with the Vivado IDE" section in the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 7].

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 developer's specific needs before instantiation through the Vivado IDE. This section provides a quick reference to parameters that can be configured at generation time.

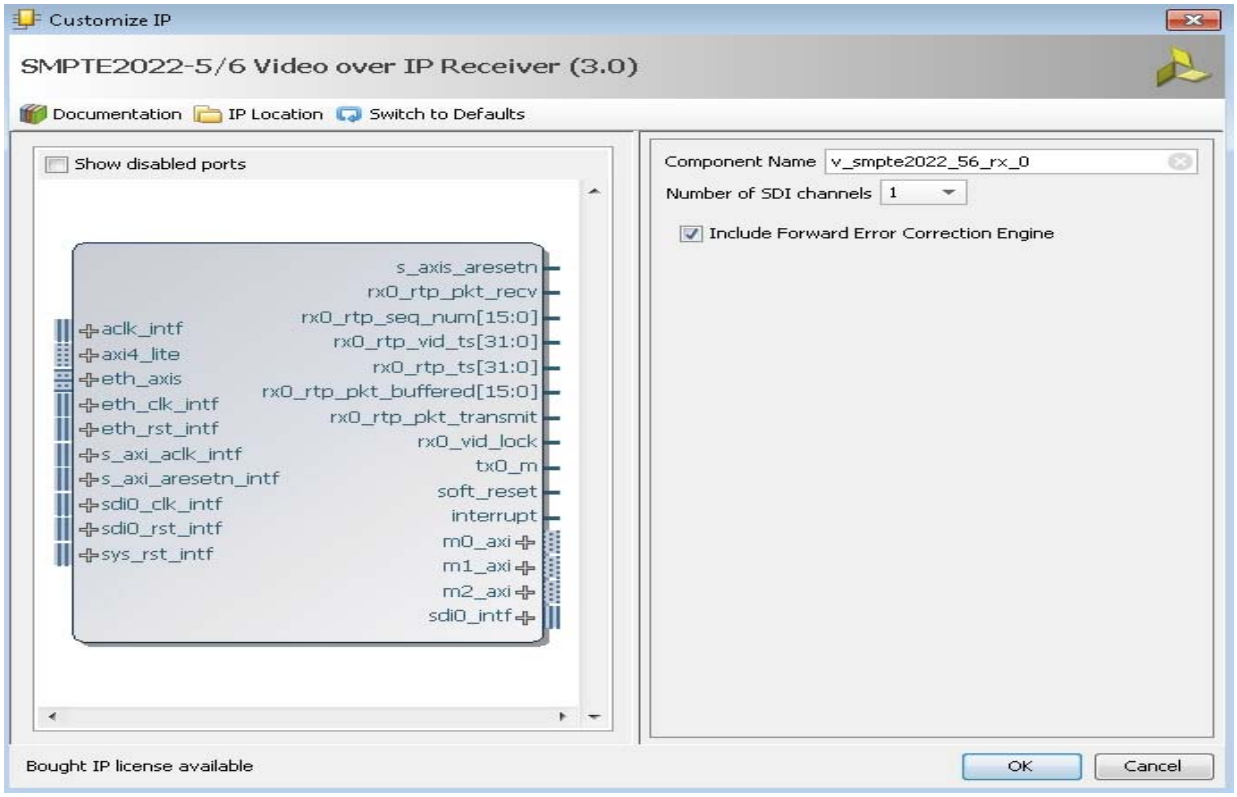

Figure 4-1: **Vivado IDE**

The Vivado IDE 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_rx 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, SMPTE 2022-5 Forward Error Correction engine is generated in the core. The core is capable of recovering IP packets lost to network transmission errors.

Output Generation

For details, see "Generating IP Output Products" in the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6].

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, Designing with the Core. 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.

Simulation

For details, see the "Simulating IP" section in the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6].

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 6].

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 Receiver Core uses an encrypted version of the VOIP Transmitter 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 Receiver core has been validated using the Xilinx Kintex®-7 FPGA Connectivity Kit. See *SMPTE 2022-5/6 High Bit Rate Media Transport over IP Networks with Forward Error Correction on Kintex-7 FPGAs* (XAPP896) [Ref 1] for more information.

Migrating and Upgrading

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 Receiver core version v3.0 has been updated to comply with the latest SMPTE 2022 5/6 Standards Specification. Migration from the older ISE version (v2.1) is supported and appropriate warning messages will be displayed during migration process.

For information about migrating to the Vivado Design Suite, see the *ISE to Vivado Design Suite Migration Guide* (UG911) [Ref 8].

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 2022-5/6 RX Core

AR 54534.

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 inserts 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 lab tools 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 9].

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. *SMPTE 2022-5/6 High Bit Rate Media Transport over IP Networks with Forward Error Correction on Kintex-7 FPGAs* (XAPP896)
2. *AXI Design 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. *Vivado Design Suite User Guide: Designing with IP* (UG896)
7. *Vivado Design Suite User Guide: Getting Started* (UG910)
8. *ISE to Vivado Design Suite Migration Guide* (UG911)
9. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
10. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)

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. Added Vivado Design Suite material and support for Virtex-7 device.
10/16/2012	2.0.1	Updated with memory requirements for the 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 Debug appendix. • Updated design to support the latest SMPTE 2022-5/6 draft change. • Removed MAC_LOW _ADDR, MAC_HIGH _ADDR, and IP_HOST_ADDR registers. • Updated screen captures in Chapter 4 and Chapter 6.
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 GUIs. • Updated Table 2-8 and 2-10.
10/02/2013	3.0	• Added XDC and module level constraints to core. • Added demonstration test bench. • Changed all signals to lowercase.

Notice of Disclaimer