LogiCORE IP SMPTE2022-5/6 Video over IP Transmitter v1.0

Product Guide

PG032 April 24, 2012

XILINX®

XILINX®

Table of Contents

Chapter 5: Constraining the Core

Chapter 6: Detailed Example Design

Appendix A: Debugging

Appendix B: Additional Resources

LogiCORE IP SMPTE2022-5/6 Video over IP Transmitter v1.0

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 6 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)	Kintex™-7, Virtex®-6
Supported User Interfaces	AXI4-Lite, AXI4-Stream, AXI4
Resources	See Table 2-1, Table 2-2, Table 2-3, and Table 2-4.
Provided with Core	
Design Files	NGC netlist, Encrypted HDL
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Not Provided
Simulation Model	VHDL or Verilog Structural
Supported S/W Driver	N/A
Tested Design Tools	
Design Entry Tools	CORE Generator™ tool
Simulation(2)	Mentor Graphics ModelSim
Synthesis Tools(2)	Xilinx Synthesis Technology (XST)
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 ISE Design Suite 14: 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 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).

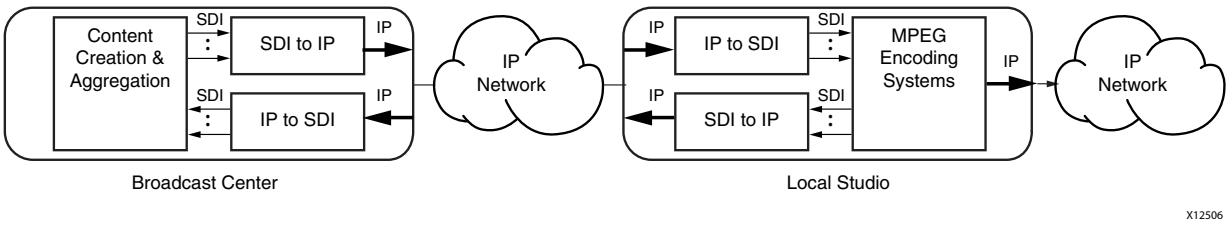

Figure 1-1: **High Bit Rate SMPTE2022-5/6 between Broadcast Center and Local Studio**

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.

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

Licensing

The core provides the following three licensing options:

- Simulation Only
- Full System Hardware Evaluation
- Full

After installing the required Xilinx software and IP Service Packs, choose a license option.

Simulation Only

The Simulation Only Evaluation license key is provided with the Xilinx® CORE Generator tool. This key lets you assess core functionality with your own design and demonstrates the various interfaces to the core in simulation. (Functional simulation is supported by a dynamically generated HDL structural model.)

No action is required to obtain the Simulation Only Evaluation license key; it is provided by default with the Xilinx CORE Generator tool.

Full System Hardware Evaluation

The Full System Hardware Evaluation license is available at no cost and lets you fully integrate the core into an FPGA design and place-and-route the design.

In addition, the license key lets you generate a bitstream from the placed and routed design, which can then be downloaded to a supported device and tested in hardware. The core can be tested in the target device for a limited time before timing out (resetting to default values and the output video becoming black), at which time it can be reactivated by reconfiguring the device.

To obtain a Full System Hardware Evaluation license, do the following:

1. Navigate to the product page for this core.
2. Click Evaluate.
3. Follow the instructions to install the required Xilinx software and IP Service Packs.

Full

The Full license key is available when you purchase the core and provides full access to all core functionality both in simulation and in hardware, including:

- Functional simulation support
- Full implementation support including place and route and bitstream generation
- Full functionality in the programmed device with no time outs

To obtain a Full license key, you must purchase a license for the core. Click the "Order" link on the Xilinx.com IP core product page for information on purchasing a license for this core.

Installing Your License File

The Simulation Only Evaluation license key is provided with the CORE Generator system and does not require installation of an additional license file. For the Full System Hardware Evaluation license and the Full license, an email will be sent to you containing instructions for installing your license file. Additional details about IP license key installation can be found in the Integrated Software Environment (ISE®) Design Suite Installation, Licensing and Release Notes document.

Ordering Information

The core is provided under the Core License Agreement and can be generated using the Xilinx CORE Generator system. The CORE Generator system is shipped with Xilinx ISE Design Suite development software.

An evaluation license for the core is shipped with the CORE Generator system. To access the full functionality of the core, including FPGA bitstream generation, a full license must be obtained from Xilinx. For more information, visit the product page.

Contact your local Xilinx sales representative for pricing and availability of additional Xilinx LogiCORE modules and software. Information about additional Xilinx LogiCORE IP modules is available on the Xilinx IP Center.

Product Specification

Standards Compliance

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) 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.

The core has three clock domains.

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

Resource Utilization

For an accurate measure of the usage of primitives, slices, and Configurable Logic Blocks (CLBs) for a particular instance, check the Display **Core Viewer after Generation** check box in the CORE Generator™ interface.

The information presented in Table 2-1, Table 2-2, Table 2-3, and Table 2-4 is a guide to the core's resource utilization for the Virtex®-6 and Kintex™-7 families.

Table 2-1: **Resource Utilization for Virtex-6 Families (FEC not included)**

No of channels	FFs	LUTs	Slices	BRAM18	BRAM36
1	9041	6953	3019	3	16
3	15317	11855	4838	5	20
6	24709	18432	8110	8	26

Table 2-2: **Resource Utilization for Virtex-6 Families (FEC included)**

No of channels	FFs	LUTs	Slices	BRAM18	BRAM36
1	11706	9453	3694	8	52
3	19151	15533	7042	10	56
6	29084	22036	11914	13	62

Table 2-3: **Resource Utilization for Kintex-7 Families (FEC not included)**

No of channels	FFs	LUTs	Slices	BRAM18	BRAM36
1	9042	6739	3218	3	16
3	15316	11975	4897	5	20
6	24767	18471	8355	8	26

Table 2-4: **Resource Utilization for Kintex-7 Families (FEC included)**

No of channels	FFs	LUTs	Slices	BRAM18	BRAM36
1	11702	9645	3965	8	52
3	18683	15191	6840	10	56
6	29066	22292	10163	13	62

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.

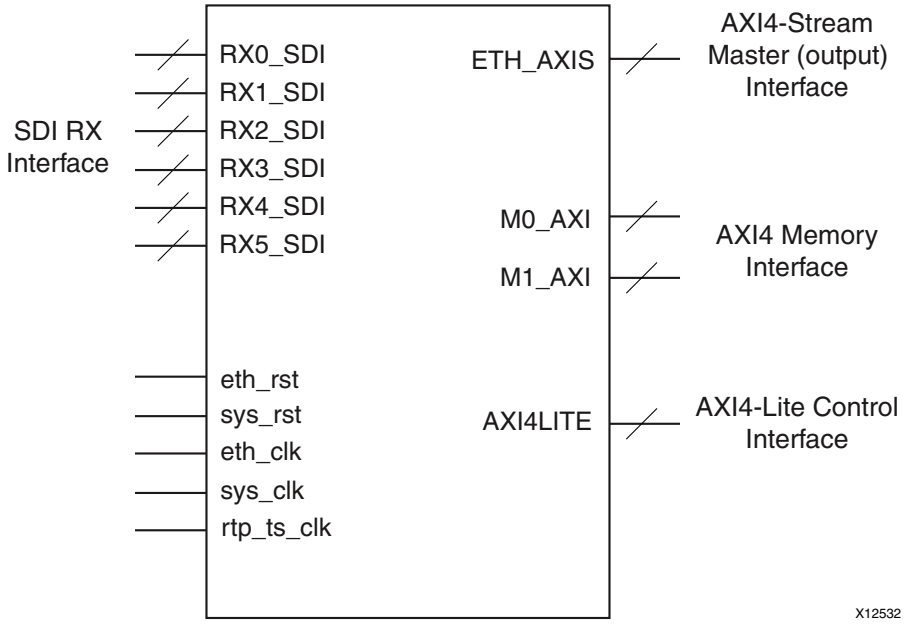

Figure 2-1: **Core Top Level Signaling Interface**

General Interface

Table 2-5 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 2-5: **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 IP Product Specification* (DS768) for more information.

Table 2-6: **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

Table 2-6: **AXI4 Memory Interface Signals *(Cont'd)***

Signal Name	Direction	Width	Description
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
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 Data
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

Table 2-6: **AXI4 Memory Interface Signals *(Cont'd)***

Signal Name	Direction	Width	Description
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 Data
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 User Guide* (UG773) for more information.

Table 2-7: **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 2-8: **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.
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 2-9: **AXI4-Lite Interface Signals**

Signal Name	Direction	Width	Description
s_axi_clk	In	1	Clock
s_axi_aresetn	In	1	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.
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

Table 2-9: **AXI4-Lite Interface Signals *(Cont'd)***

Signal Name	Direction	Width	Description
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's 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) BASEADDR +	Register Name	Access Type	Default Value	Register Description
General Registers				
0x0000	CONTOL	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	0x01000000	7-0: REVISION_NUMBER 11-8: PATCH_ID 15-12: VERSION_REVISION 23-16: VERSION_MINOR 31-24:VERSION_MAJOR
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
0x0084	HDR_PARAM	R/W	0	Bit 0: Reserved Bit 1: Include Video Timestamp 31-2: Reserved
0x00A0	NUM_CHANNEL	R	0	10-0: Number of channels in design 31-11: reserved

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

Address (hex) BASEADDR +	Register Name	Access Type	Default Value	Register Description
Channel Registers				
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_L 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)

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 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 facilitate software reset. When '1' all registers and the core is held at reset.

CHANNEL (0x0030) Register

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

Version (0x0010) 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.

NUM_CHANNEL (0x00A0) Register

This register indicates the number of channels in design.

FEC_CONFIG (0x0104) Register

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

FEC_OFFSET (0x0108) Register

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

FEC_L (0x010C) Register

The FEC_L register is for configuring L value of FEC matrix.

Level A FEC 1 <= L<= 1020 and

Level B FEC 4 <= L <= 1020.

FEC_D (0x0110) Register

The FEC_D register is for configuring D value of 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.

Customizing and Generating the Core

This chapter includes information on using Xilinx tools to customize and generate the core.

Graphical User Interface

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.

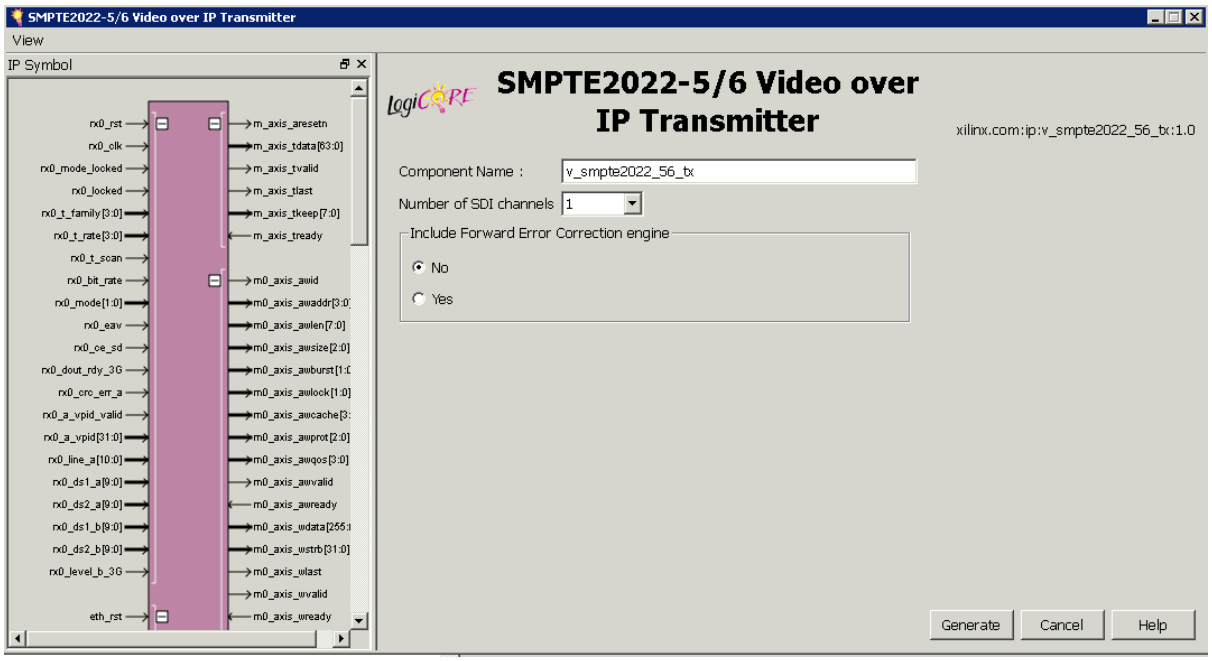

Figure 3-1: **SMPTE2022-5/6 Video over IP Transmitter Coregen Graphical User Interface**

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_rx_v1_0 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 Froward Error Correction engine.

Output Generation

The Xilinx® CORE Generator system 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 the following files.

Table 3-1: **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 software designs.

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

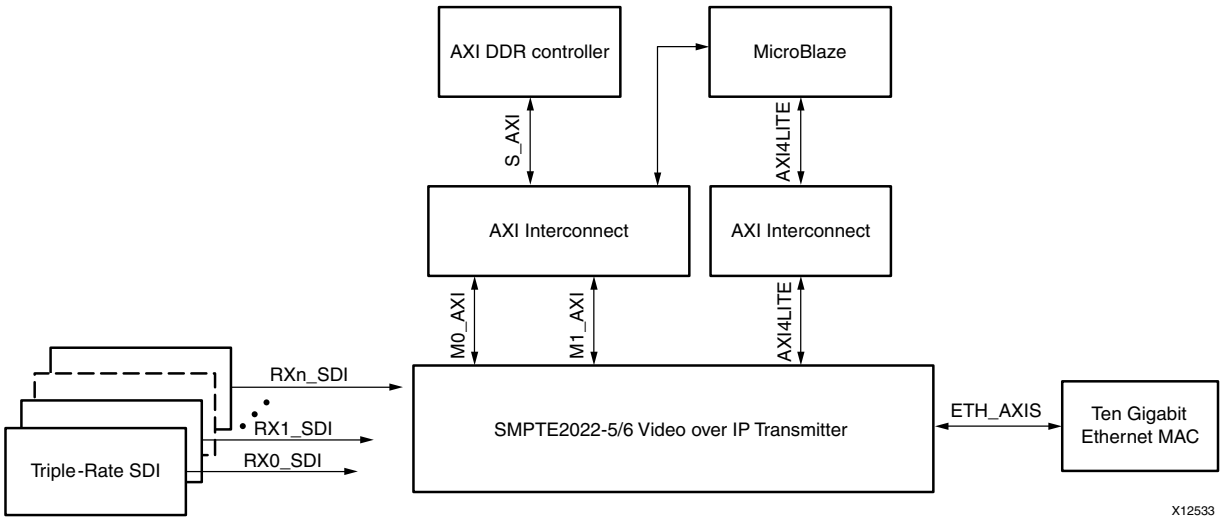

Figure 4-1: **SMPTE2022-5/6 Video over IP Transmitter System Built with other Xilinx IP Cores**

Constraining the Core

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

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

Appendix A

Debugging

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

Appendix B

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.

Solution Centers

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

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.

Notice of Disclaimer