Video over IP FEC Transmitter v2.0

LogiCORE IP Product Guide

Vivado Design Suite

PG206 October 5, 2016

XILINX
ALL PROGRAMMABLE™

Table of Contents

Appendix D: Additional Resources and Legal Notices and Legal Notices

Introduction

The Xilinx® LogiCORE™ IP Video over IP FEC Transmitter is a broadcast application module that transmits additional redundant packets along with the original incoming packets to help the receiver recover the possible loss of packets due to a network error. The redundant packets are Real time Transport Protocol (RTP) encapsulated payloads using SMPTE ST 2022-1 and SMPTE ST 2022-5 Forward Error Correction (FEC) method. It is capable of handling number of video streams and is suited for deploying SMPTE ST 2022-1 stream in 1 Gb/s networks and SMPTE ST 2022-5 streams in 10 Gb/s networks. This core is used for developing Internet Protocol-based systems that reduce the overall cost of distribution and routing of audio and video data.

Features

- SMPTE ST 2022-1 and SMPTE ST 2022-5 based FEC encoding
- Up to 512 channels for SMPTE ST 2022-1 based FEC encoding scheme and up to 8 channels for SMPTE ST 2022-5 based FEC encoding scheme (configurable at compilation time)
- FEC matrix selection per channel
    - SMPTE ST 2022-1 FEC matrix selection per channel
        - 1 <= FEC L <= 20
        - 4 <= FEC D <= 20
        - FEC L * FEC D <= 100

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	UltraScale+™, UltraScale, Zynq®-7000, Virtex®-7, Kintex®-7
Supported User Interfaces	AXI4-Lite, AXI4-Stream, AXI-4
Resources	See Resource Utilization.
Provided with Core	
Design Files	Encrypted HDL
Example Design	N/A
Test Bench	Verilog
Constraints File	XDC
Simulation Model	Encrypted RTL
Supported S/W Driver(2)	N/A
Tested Design Flows(3)	
Design Entry	Vivado® Design Suite Vivado
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide.
Synthesis	Vivado Synthesis
Support	
Provided by Xilinx at the Xilinx Support web page	

Notes:

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

Features continued

- SMPTE ST 2022-5 FEC matrix selection per channel
  - 1 <= FEC L <= 1020
  - 4 <= FEC D <= 255
  - SD-SDI stream
    - FEC L * FEC D <= 1500
  - HD-SDI stream
    - FEC L * FEC D <= 3000
  - 3G-SDI stream
    - FEC L * FEC D <= 6000
- FEC interleaving selection per channel
  - Block aligned
  - Non block aligned
- FEC operating mode configurable per channel
  - Packet bypass (No FEC)
  - 1D FEC (Column)
  - 2D FEC (Column and Row)

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 Ethernet for the transmission of multiple compressed media streams is becoming a major customer requirement. The industry is primarily looking at the SMPTE ST 2022 set of standards to create an open and interoperable way of connecting video over Ethernet 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, Video over IP FEC is currently aimed at multiple SMPTE ST 2022-1 streams carried on 1GbE networks and SMPTE ST 2022-5 streams carried on 10GbE networks.

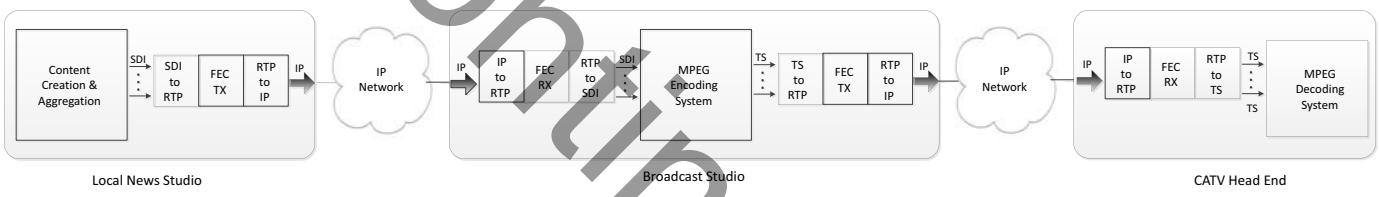


Figure 1-1: **Video over IP FEC Usage in Contribution or Distribution Networks**

The core includes Forward Error Correction (FEC). FEC protects the transport streams during the transport over IP networks. With FEC, the transmitter adds systematically generated redundant data. 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 providing a modular device with highly integrable 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 1 Gbps or 10 Gbps Ethernet, 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 SMPTE ST 2022-1/2 and SMPTE ST 2022-5/6 supported Video over IP FEC Transmitter core generates the Forward Error Correction packets in accordance with SMPTE ST 2022-1 and SMPTE ST 2022-5 for recovery of IP packets lost due to network transmission errors. The core supports up to 8 channel in SMPTE ST 2022-5/6 operating mode and up to 512 channels in SMPTE ST 2022-1/2 operating mode. The core support multiple FEC mode operation depends on input packet type and user preferences. The core has symmetric and well-defined input and output payload interface which following same AXI4-Stream protocol.

Applications

The SMPTE ST 2022-1 or SMPTE ST 2022-5 supported Video over IP FEC Transmitter core is used to transport compressed and uncompressed constant bit rate video streams over an IP network.

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. 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 Video over IP FEC Transmitter v2.0 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.

License Checkers

If the IP requires a license key, the key must be verified. The Vivado® design tools have several license checkpoints 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 tools:

- 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.*

Product Specification

Architecture Overview

Figure 2-1 shows the Video over IP FEC Transmitter core architecture.

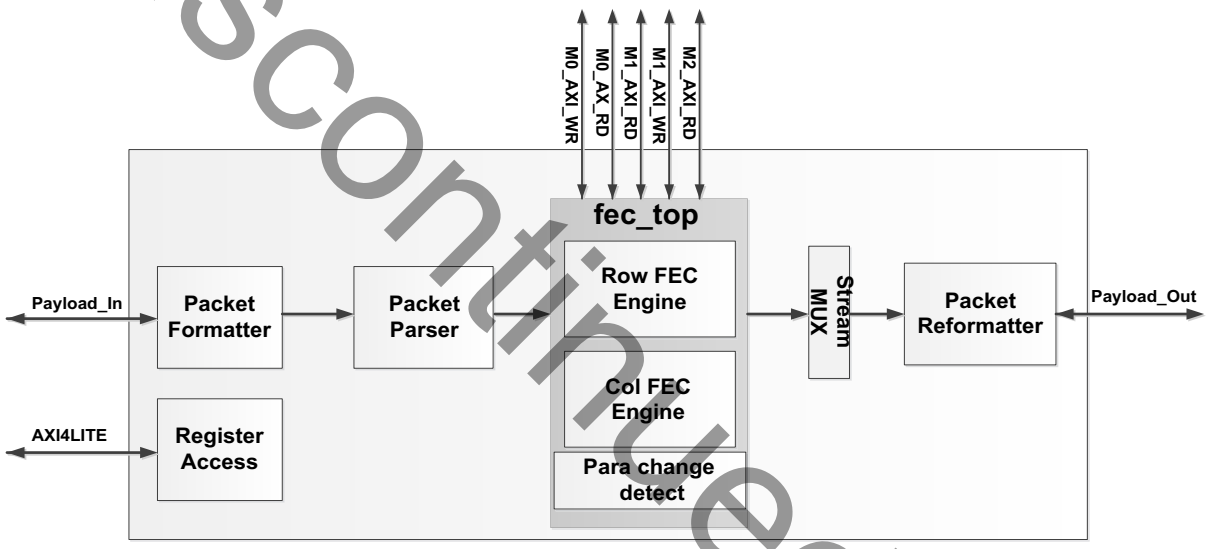


Figure 2-1: **Architecture Overview of Video over IP FEC Transmitter**

The main functional blocks of the core are:

Packet Formatter: Extract packet data and side-band info from AXI4-Stream payload in interface, and store packet data and meta-data in Block RAMs.

Packet Parser: Coordinate row and column FEC engine to process incoming RTP packets. At same time, it passes the RTP packets to stream Multiplex to be sent out.

Row FEC Engine: Do row FEC calculation and prepare row FEC packets. It store intermediate packet data in DDR.

Col FEC Engine: Do column FEC calculation and prepare column FEC packets. It store intermediate packet data in DDR.

Para Change Detect: Detects channel parameter change (FEC L, FEC D, Interleave, payload length, FEC mode). When any of these parameter change, the FEC engine stops the current

FEC calculation and completely restarts a new matrix based on new parameters from next RTP packet. Note, parameter change does not cause RTP packet corruption but stops the current matrix calculation. (Note, the RTP packet after parameter change is skipped for FEC calculation).

Stream MUX: Multiplex the three packet streams to be sent out: RTP packets, column and row FEC packets.

Packet Reformatter: Inverse function of packet formatter. Prepares AXI4-Stream payload out protocol based on packet data and side band info from FEC engine.

Register Access: Register configuration and status read-back on the core.

Standards

The SMPTE ST 2022-1 and SMPTE ST 2022-5 supported Video over IP FEC Transmitter core is compliant with the AXI4, AXI4-Stream, and AXI4-Lite interconnect standards. See the Video IP: AXI Feature Adoption section of the Vivado AXI Reference Guide (UG1037) [Ref 1] for additional information.

Performance

Maximum Frequencies

The maximum achievable clock frequency and all resource counts can be affected by other tool options, additional logic in the FPGA device, different versions of Xilinx® tools, and other factors.

Resource Utilization

For details about resource utilization, visit Performance and Resource Utilization.

Resource Utilization is calculated using `core_clk` and the frequency of other clock fixed at `s_axi_aclk`=100 MHz. The maximum clock frequency results were obtained by double-registering input and output ports to reduce dependence on I/O placement. The inner level of registers used a separate clock signal to measure the path from the input registers to the first output register through the core. The results are post-implementation, using tool default settings except for high effort.

The resource usage results do not include the "characterization" registers and represent the true logic used by the core. LUT counts include SRL16s or SRL32s.

Clock frequency does not take clock jitter into account and should be de rated by an amount appropriate to the clock source jitter specification. The maximum achievable clock frequency and the resource counts might also be affected by other tool options, additional logic in the FPGA, using a different version of Xilinx tools, and other factors.

Port Descriptions

The SMPTE ST 2022-1 or SMPTE ST 2022-5 supported Video over IP FEC Transmitter 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 provides an I/O diagram of the core.

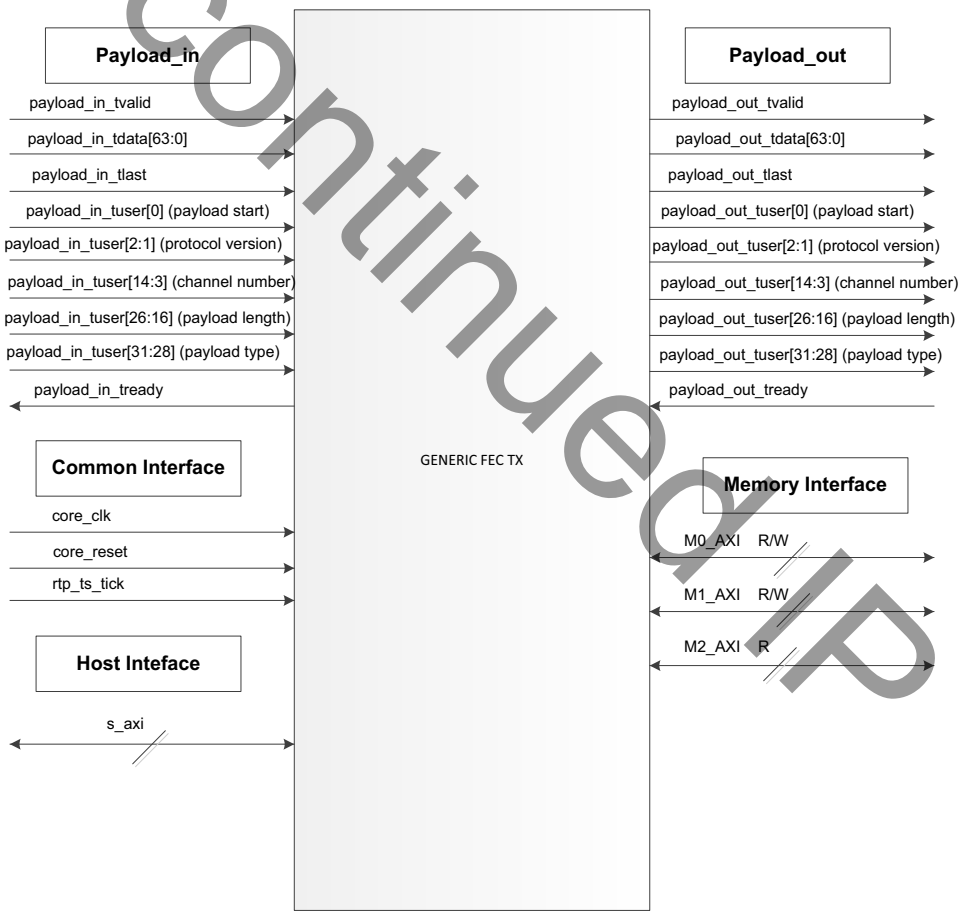


Figure 2-2: **Core I/Os**

Common Interface

Table 2-1 describes the common interface signals.

Table 2-1: **Common Interface Signals**

Signal Name	Direction	Width	Description
core_clock	In	1	Main operating clock for the core. It applies to all the interfaces except the host interface.
core_reset	In	1	Main reset for the core. It applies to payload_in and payload_out interfaces. It is synchronous to core_clock and is Active High.
rtp_ts_tick	In	1	Data Pulse Input which is synced to core_clk to generate ST 2022-1 or ST 2022-5 RTP timestamp. The frequency of the pulse is 90 kHz for ST 2022-1/2 mode and 27 MHz for STm 2022-5/6 mode.

Stream Payload In Interface (payload_in)

Table 2-2 describes the stream payload in interface signals.

Table 2-2: **Transmitter Stream Payload In Signals**

Signals	Direction	Width	Description
payload_in_tvalid	In	1	High only from the start to the end of the packet transfer.
payload_in_tdata	In	64	Packet Data
payload_in_tlast	In	1	High only at the last word of the output packet

Table 2-2: **Transmitter Stream Payload In Signals *(Cont'd)***

Signals	Direction	Width	Description		
payload_in_tuser	In	32	Bit	Abbreviation	Description
			0	Packet Start	High only at the first valid word of the input packet.
			2:1	Protocol Version	Protocol version (set to "00")
			14:3	Channel Number	Shall be valid at packet start
			15	Reserved	
			26:16	Packet Length	Shall be valid at packet start. It is the sum of packet length in bytes.
			27	Reserved	
			31:28	Packet Type	Shall be valid at packet start
					Value: 0000 — UDP encapsulated
					Value: 0001 — RTP encapsulated ST2022-2 compliant media packet
					Value: 0101 — RTP encapsulated ST2022-6 compliant media packet
					Value: 1000 — RTP encapsulated RFC4175 compliant media packet
					Value: 1001 — RTP encapsulated RFC3190 compliant media packet
payload_in_tready	Out	1	Asserted low in between packet transfers.		

Note: First packet byte is in bit[7:0] of the first word transfer. Able to accept up to 2KB per packet transfer.

Stream Payload Out Interface (payload_out)

Table 2-3 describes the stream payload out interface signals.

Table 2-3: **Transmitter Stream Payload Out Signals**

Signals	Direction	Width	Description
payload_out_tvalid	Out	1	High only from the start to the end of the packet transfer.
payload_out_tdata	Out	1	Packet Data
payload_out_tlast	Out	1	High only at the last word of the output packet

Table 2-3: **Transmitter Stream Payload Out Signals *(Cont'd)***

Signals	Direction	Width	Description			
payload_out_tuser	Out	32	Bit	Abbreviation	Description	
			0	Packet Start	High only at the first valid word of the output packet.	
			2:1	Protocol Version	Protocol version (set to "00")	
			14:3	Channel Number	Shall be valid at packet start	
			15	Reserved		
			26:16	Packet Length	Shall be valid at packet start. It is the sum of packet length in bytes.	
			27	Reserved		
			31:28	Packet Type	Shall be valid at packet start	
					Values	Description
					0000	UDP encapsulated
					0001	RTP encapsulated ST2022-2 compliant media packet
					0010	RTP encapsulated ST2022-1 compliant Column FEC
					0011	RTP encapsulated ST2022-1 compliant Row FEC packet
					0101	RTP encapsulated ST2022-6 compliant media packet
					0110	RTP encapsulated ST2022-5 compliant Column packet
					0111	RTP encapsulated ST2022-5 compliant Row FEC packet
					1000	RTP encapsulated RFC4175 compliant media packet
					1001	RTP encapsulated RFC3190 compliant media packet
payload_out_tready	In	1	Asserted low in between packet transfers.			

Note: First packet byte is in bit[7:0] of the first word transfer. Able to transfer up to 2KB per packet.

Figure 2-3 shows a waveform diagram of the stream payload protocol.

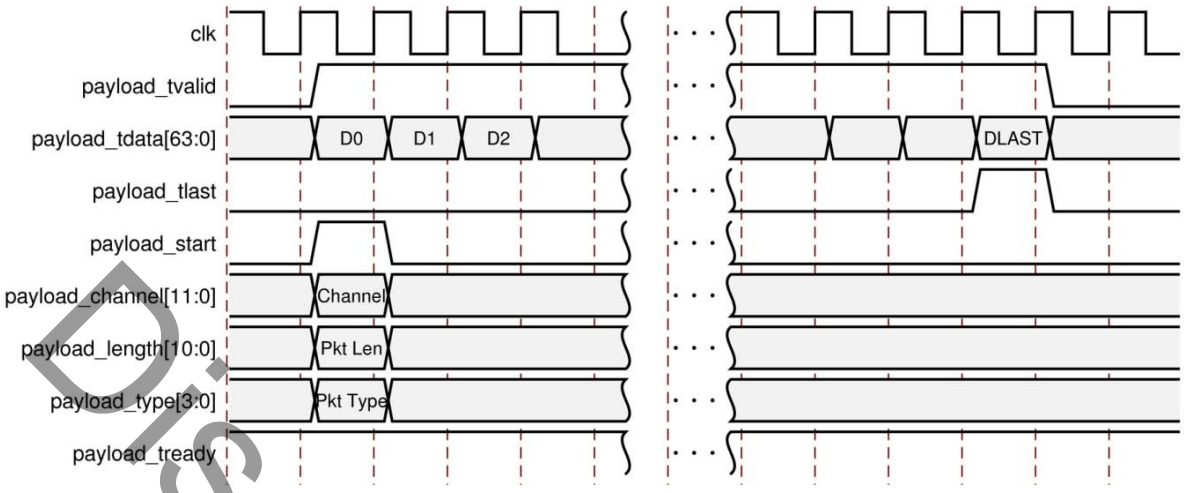


Figure 2-3: **Stream Payload Protocol Waveform**

Host Interface (S_AXI)

Table 2-4 describes the host interface signals. Refer to *AXI Reference Guide* (UG1037) [Ref 7] on AXI-Lite interface and its protocol.

Table 2-4: **Host 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	32	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.

Table 2-4: **Host Interface Signals *(Cont'd)***

Signal Name	Direction	Width	Description
s_axi_arready	Out	1	Ready. Indcicates DMA is ready to accept the read address.
s_axi_araddr	In	32	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.
Table 4. AXI4-Lite Interface Signals			

Memory Interface (M*_AXI_*)

See the *AXI Reference Guide* (UG1037) [Ref 7] for more information on AXI-MM interface and its protocol. The AXI-MM port data width is 256.

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

Signal Name	Direction	Width	Description
M0_AXI_WR: COLUMN FEC DDR Write: Intermediate Result to DDR			
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_awcache	Out	4	Write Address Channel Cache Characteristics.
m0_axi_awprot	Out	3	Write Address Channel Protection Bits.
m0_axi_awaddr	Out	32	Write Address Channel Address.
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_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_awid	Out	1	Write Address Channel Transaction ID.

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

Signal Name	Direction	Width	Description
m0_axi_awlock	Out	2	Write Data Channel Atomic Access Type.
m0_axi_awqos	Out	4	Write Data Channel Quality of Service.
m0_axi_bid	In	1	Write Response Channel Transaction ID.
M0_AXI_RD: COLUMN FEC DDR Read: preload from DDR			
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_arcache	Out	4	Read Address Channel Cache Characteristics.
m0_axi_arprot	Out	3	Read Address Channel Protection Bits.
m0_axi_araddr	Out	32	Read Address Channel Address.
m0_axi_arvalid	Out	1	Read Address Channel Valid.
m0_axi_arready	In	1	Read Address Channel Ready.
m0_axi_rready	Out	1	Read Data Channel Ready.
m0_axi_rdata	In	256	Read Data Channel Data.
m0_axi_rlast	In	1	Read Data Channel Last Data Beat.
m0_axi_rvalid	In	1	Read Data Channel Valid.
m0_axi_rresp	In	2	Read Data Channel Response Code.
m0_axi_arid	Out	1	Read Address Channel Transaction ID.
m0_axi_arlock	Out	2	Read Address Channel Atomic Access Type.
m0_axi_arqos	Out	4	Read Address Channel Channel Quality of Service.
m0_axi_rid	In	1	Read Data Channel Transaction ID.
M1_AXI_RD: COLUMN FEC DDR Read: result FEC packet from DDR			
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_arcache	Out	4	Read Address Channel Cache Characteristics.
m1_axi_arprot	Out	3	Read Address Channel Protection Bits.
m1_axi_araddr	Out	32	Read Address Channel Address.
m1_axi_arvalid	Out	1	Read Address Channel Valid.
m1_axi_arready	In	1	Read Address Channel Ready.
m1_axi_rready	Out	1	Read Data Channel Ready.
m1_axi_rdata	In	256	Read Data Channel Data.
m1_axi_rlast	In	1	Read Data Channel Last Data Beat.

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

Signal Name	Direction	Width	Description
m1_axi_rvalid	In	1	Read Data Channel Valid.
m1_axi_rresp	In	2	Read Data Channel Response Code.
m1_axi_arid	Out	1	Read Address Channel Transaction ID.
m1_axi_arlock	Out	2	Read Address Channel Atomic Access Type.
m1_axi_arqos	Out	4	Read Address Channel Channel Quality of Service.
m1_axi_rid	In	1	Read Data Channel Transaction ID.
M1_AXI_WR: ROW FEC DDR Write: intermediate result to DDR			
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_awcache	Out	4	Write Address Channel Cache Characteristics.
m1_axi_awprot	Out	3	Write Address Channel Protection Bits.
m1_axi_awaddr	Out	32	Write Address Channel Address.
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_bresp	In	2	Write Response Channel Response Code.
m1_axi_bvalid	In	1	Write Response Channel Valid.
m1_axi_bready	Out	1	Write Response Channel Ready.
m1_axi_awid	Out	1	Write Address Channel Transaction ID.
m1_axi_awlock	Out	2	Write Data Channel Atomic Access Type.
m1_axi_awqos	Out	4	Write Data Channel Quality of Service.
m1_axi_bid	In	1	Write Response Channel Transaction ID.
M2_AXI_RD: ROW FEC DDR Read: preload from DDR			
m2_axi_arlen	Out	8	Read Address Channel Burst Length code.
m2_axi_arsize	Out	3	Read Address Channel Transfer Size code.
m2_axi_arburst	Out	2	Read Address Channel Burst Type.
m2_axi_arcache	Out	4	Read Address Channel Cache Characteristics.
m2_axi_arprot	Out	3	Read Address Channel Protection Bits.

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

Signal Name	Direction	Width	Description
m2_axi_araddr	Out	32	Read Address Channel Address.
m2_axi_arvalid	Out	1	Read Address Channel Valid.
m2_axi_arready	In	1	Read Address Channel Ready.
m2_axi_rready	Out	1	Read Data Channel Ready.
m2_axi_rdata	In	256	Read Data Channel Data.
m2_axi_rlast	In	1	Read Data Channel Last Data Beat.
m2_axi_rvalid	In	1	Read Data Channel Valid.
m2_axi_rresp	In	2	Read Data Channel Response Code.
m2_axi_arid	Out	1	Read Address Channel Transaction ID.
m2_axi_arlock	Out	2	Read Address Channel Atomic Access Type.
m2_axi_arqos	Out	4	Read Address Channel Channel Quality of Service.
m2_axi_rid	In	1	Read Data Channel Transaction ID.

Note: The ROW FEC result is not written into DDR and is different from COL FEC result.

Register Space

The Video over IP FEC Transmitter register space is partitioned to general and channel-specific registers.

Table 2-6: **Host Interface (S_AXI [AXI4-Lite]) Register Map**

Address Offset (HEX)	Register Name	Access Type	Cleared with SOFT reset	Default Value (HEX)	Description	
					Bit Range	Value
General						
0x0000	control	R/W	N	0x00000000	Control	
					31:2	Reserved
					1	Channel Update
						Send pulse to allow configured registers to take effect for the channel.
					0	Soft reset
						1 Reset all VoIP FEC Transmitter Registers
						0 Un-Reset all VoIP FEC Transmitter Registers
0x0004	status	R	N	0x00000000	Status	
					31:1	Reserved
					0	Update Busy
						1 Updating configured parameters to the core
						0 No updating
0x0008	channel_access	R/W	Y	0x00000000	Channel Access	
					31:12	Reserved
					11:0	The channel number to access registers

Table 2-6: **Host Interface (S_AXI [AXI4-Lite]) Register Map *(Cont'd)***

Address Offset (HEX)	Register Name	Access Type	Cleared with SOFT reset	Default Value (HEX)	Description	
					Bit Range	Value
0x000C	sys_config	R	N	0x00000000	System Configuration	
					31:26	Reserved
					25:16	Maximum Supported FEC L (Column FEC)
					20	Maximum FEC L for ST2022-1/2
					1020	Maximum FEC L for ST2022-5/6
					15:12	Reserved
					11:0	Number of Maximum Channels Supported
0x0010	version	R	N	0x02000000	Hardware Version	
					31:24	Version major
					23:16	Version minor
					15:12	Version revision
					11:8	Patch ID
					7:0	Revision number
0x0018	fec_base_address	R/W	Y	C_FEC_BASEADDRESS	FEC Base Address	
					31:0	Base Address of buffer allocated in the external memory (via AXI-MM Interface). Refer to Memory Requirement in Chapter 3 for more information.
0x0024	in_pkt_cnt	R	Y	0x00000000	Received Packet Count	
					31:0	Incoming packet count at the payload in interface of the core

Table 2-6: **Host Interface (S_AXI [AXI4-Lite]) Register Map *(Cont'd)***

Address Offset (HEX)	Register Name	Access Type	Cleared with SOFT reset	Default Value (HEX)	Description	
					Bit Range	Value
0x0028	out_pkt_cnt	R	Y	0x00000000	Transmit Packet Count	
					31:0	Transmitted packet count at the payload out interface of the core.
Channel						
0x0080	fec_config	R/W	Y	0x00000000	FEC Configuration	
					31:21	Reserved
					20	FEC Interleaving
						0 Block Align
						1 Non-Block Align
					19:18	FEC Mode
						00 Bypass Mode
						01 Reserved
						10 FEC 1D
						11 FEC 2D
					17:8	FEC L
					7:0	FEC D

The registers are categorized into two sections: the general space and channel space. The registers in the general space apply to all the channels while the channel space registers apply to the specific channel set by register offset 0x08.

The general registers can be access through normal address read and write.

To configure a channel, observe the following steps.

1. Before configuring a channel, check the busy bit (bit 0, register offset 0x04) to be Low.
2. Set the channel to be configured at register offset, 0x08.
3. Configure the channel specific registers.
4. Pulse channel update bit (bit 1, register offset 0x00) to commit the channel parameters.
5. Repeat steps 1-4 for each channel.

See Figure 2-4. To read off a channel register, set the channel access at offset 0x08 before proceeding.

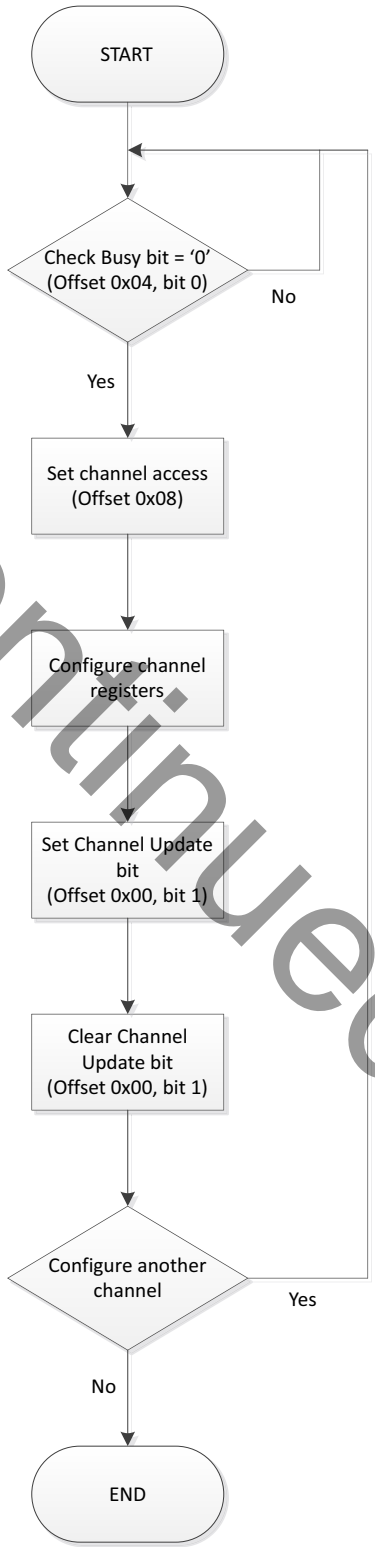


Figure 2-4: **Flow Chart**

CONTROL (0x0000) Register

Used to generate an update pulse to make channel configuration take effect. Only bit 1 is used. Program 1 to bit 1 then program 0 to bit 1 to generate the update pulse after each channel configuration. it 0 is used to generate soft reset to reset register space and does not reset the core logic.

STATUS (0x0004) Register

Since channel update process takes multiple clock cycles, user can move to next channel programming only when current channel configuration is done, which is indicated by bit 0 of STATUS register (0 means updating done).

CHANNEL_ACCESS (0x0008) Register

12-bit register bit used to select which channel to be configured. For each channel, it has its own copy of channel registers (offset address 0x0080). This register is used as page address used to access each channel's (each page) channel registers.

SYS_CONFIG (0x000C) Register

This register reflects maximum supported matrix FEC L value, for SMPTE ST 2022-1/2 it is 20 and SMPTE ST 2022-5/6 it is 1020 and the value of core generic settings during compilation time "C_CHANNELS", namely the maximum number of channels supported by current core configuration.

VERSION (0x0010) Register

Version of the core design. Such as: initial version number is 0x01000000, namely version 1.0.

FEC_BASE_ADDRESS (0x0018) Register

The base address of DDR which is allocated for this core to operate normally. For 512 channel case, the DDR address space allocated for care is base address ~ base address + 64 MBytes.

IN_PKT_CNT (0x0024) Register

This register records how many RTP packets received by the core which come from the AIXS payload in interface.

OUT_PKT_CNT (0x0028) Register

This register records how many RTP and FEC packets are available at output from the core, and are all sent out at AXI4-Stream payload out interface.

FEC_CONFIG (0x0080) Register

Note, this is channel space register. Each channel have one copy of this register. To configure a channel, you must program the `CHANNEL_ACCESS` register with channel number first. Then configure channel info to this `FEC_CONFIG` register.

Note: bit19-bit18 is FEC Mode. "11" means 2D (both COLUMN and ROW FEC packets are generated).

"10" means 1D (only COLUMN FEC packets are generated). "00" means bypass (no FEC packets are generated).

Any one of below configuration is considered a user error and the core is forced into "bypass" mode. In bypass mode, there are no FEC packets generated but no influence on RTP packets at core output.

1. FEC D < 4.
2. FEC L < 4 and FEC Mode = "11". (For 2D, minimum matrix size is 4x4).
3. FEC L=1 and Non block align.

Designing with the Core

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

General Design Guidelines

Figure 3-1 shows an example of an application design using Video over IP FEC TX core with other Xilinx IP.

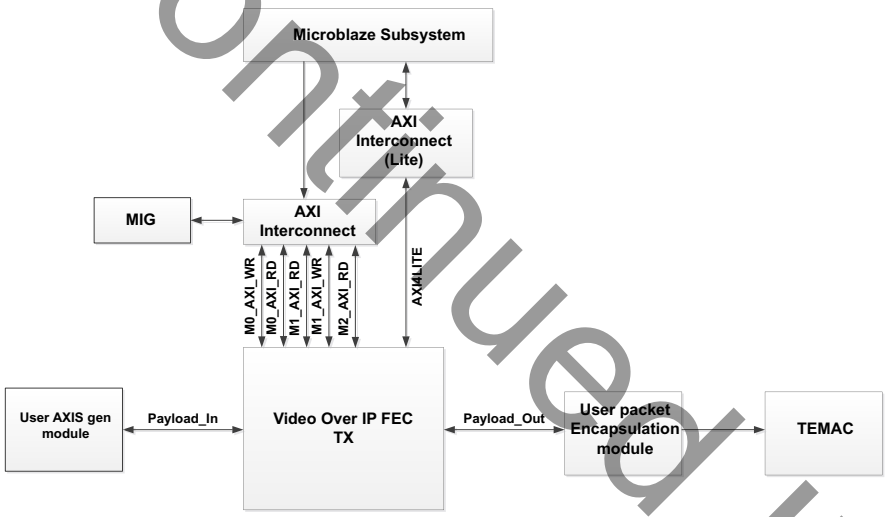


Figure 3-1: **Example of an Application Design with Video over IP FEC TX Core**

This section describes how Video over IP FEC TX core can be designed to build a fully functional design with user application logic.

The core accepts RTP encapsulated SMPTE ST 2022-2 or SMPTE ST 2022-6 compliant media packet for ST 2022-1/5 FEC packet generation. Figure 3-2 and Figure 3-4 shows the packet structure for incoming data format to VoIP FEC Transmitter core. The incoming RTP encapsulated ST 2022 media packet and the `payload_in` TUSER must be compliant to the stream payload in interface protocol in Table 2-2.

The output of the core are RTP encapsulated ST 2022-2/6 media packets or RTP encapsulated ST 2022-1/5 FEC row or ST2022-1/5 FEC column packet based on

`fec_config` (0x80) settings. Figure 3-3 and Figure 3-5 shows the packet structure for FEC packets.

For incoming SMPTE 2022-6 RTP packets, the `fec_usage` field of the media payload header is modified according to FEC mode programmed by user before the packet is sent out at output of core or is used for FEC packet generation.

In the Video over IP FEC TX core, a few parameters are defined: fec_mode, L, D, interlave, and packet_length. In an FEC operation, these parameters are assumed to be static. If any of these five parameters changes during normal operation, the core detects the change as a parameter change. Upon detecting the parameter change, the core starts a new FEC process from a new matrix starting point. This feature is for dynamic parameter change support.

Note that, in RTP header, the header extension field should be fixed at zero because header extension is not supported.

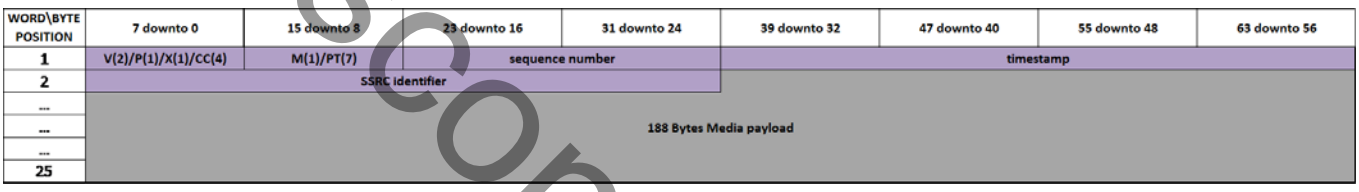


Figure 3-2: **Media Payload with SMPTE ST 2022-2 Header**

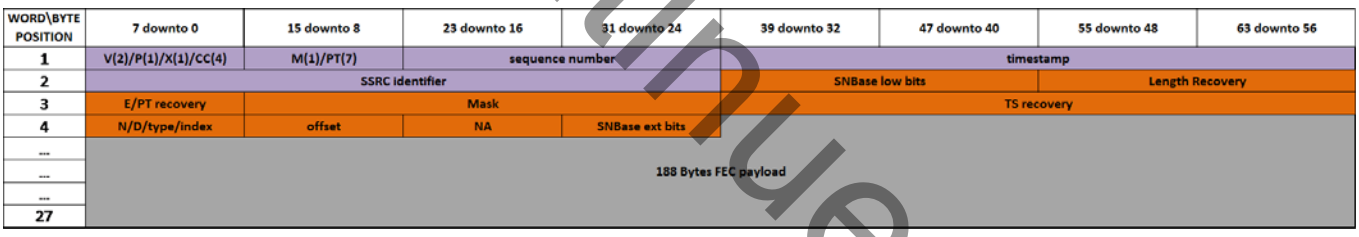


Figure 3-3: **FEC Payload with SMPTE ST 2022-1 Header**

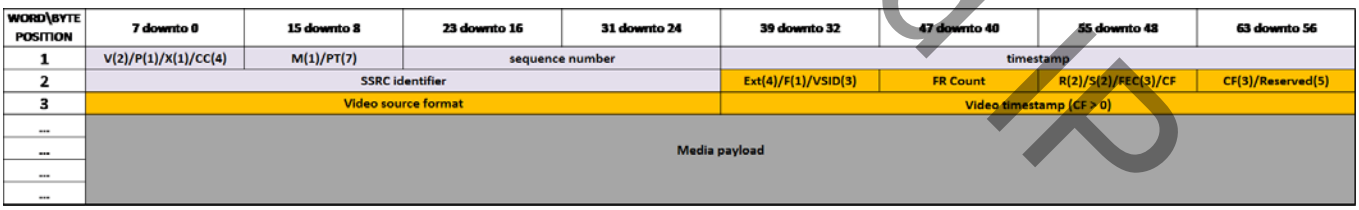


Figure 3-4: **Media Payload with SMPTE ST 2022-6 Header**

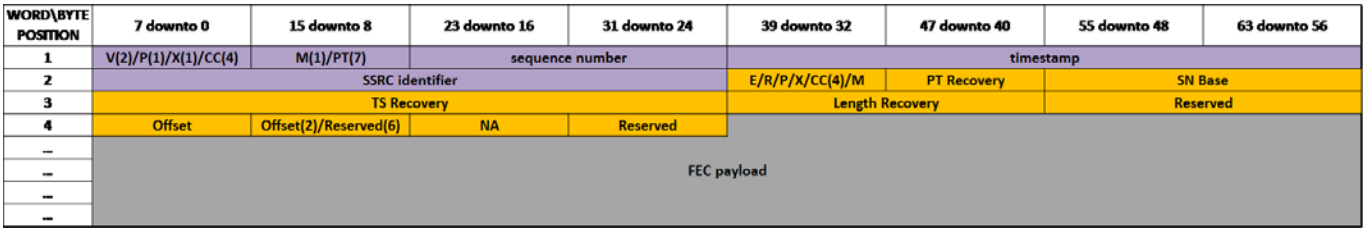


Figure 3-5: **FEC Payload with SMPTE ST 2022-5 Header**

Memory Requirement

The Video over IP FEC Transmitter core requires access to the external memory for normal operation and to perform FEC generation. The core stores intermediate ST 2022-1/5 FEC packets (column/row) and result FEC (column) packets at different sections of the external memory. The AXI-MM data port width supported is 256. Set the FEC Base Address in the external memory for whole VoIP FEC Transmitter core usage. For each channel, a fixed memory address range is allocated. In this range, it's further divided into multiple slots which is marked by indexes. Each slot is fixed at 2KB which is used to store one packet.

The size allocated in the external memory for each packet is 2KB. The Video over IP FEC Transmitter core utilizes 64 packet slots per channel in SMPTE ST 2022-1/2 operating mode and 2048 packet slots per channel in SMPTE ST 2022-5/6 operating mode. DDR memory requirement in SMPTE ST 2022-1/2 operating mode for 512 channels is 64MB and in SMPTE ST 2022-5/6 operating mode for 8 channels is 32 MB.

Clocking

The Video over IP FEC Transmitter core has only one clock domain:

- Core clock domain, `core_clk` (Fmin for `core_clk` is arbitrarily set at 100 Mhz. This is based on a calculated assumption that the core is able to handle an input bandwidth of 1G at this frequency). For 10 G operation, the `core_clk` is suggested to be set to 200 Mhz. All the AXI4-stream and AXI4 interfaces, along with the main core activities are under this clock.
- AXI4-Lite clock domain, `s_axi_aclk` (Fmin for `s_axi_aclk` is set at 50 MHz). Core register access works with this clock.

Resets

The Video over IP FEC Transmitter core has two reset input ports:

- Core domain reset, `core_reset`.
- AXI4-Lite domain reset, `s_axi_aresetn`

The two resets from input port must be synchronous to their individual clock domains. A minimum of 16 clock cycles is recommended for the reset assertion. Reset signal `s_axi_aresetn` is to be de-asserted last.

Design Flow Steps

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

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 2]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 8]
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 9]

Customizing and Generating the Core

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

If you are customizing and generating the core in the Vivado IP Integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 2] for detailed information. IP Integrator does not auto-compute any configuration value for this core.

Vivado Integrated Design Environment

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

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

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6] and the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 8].

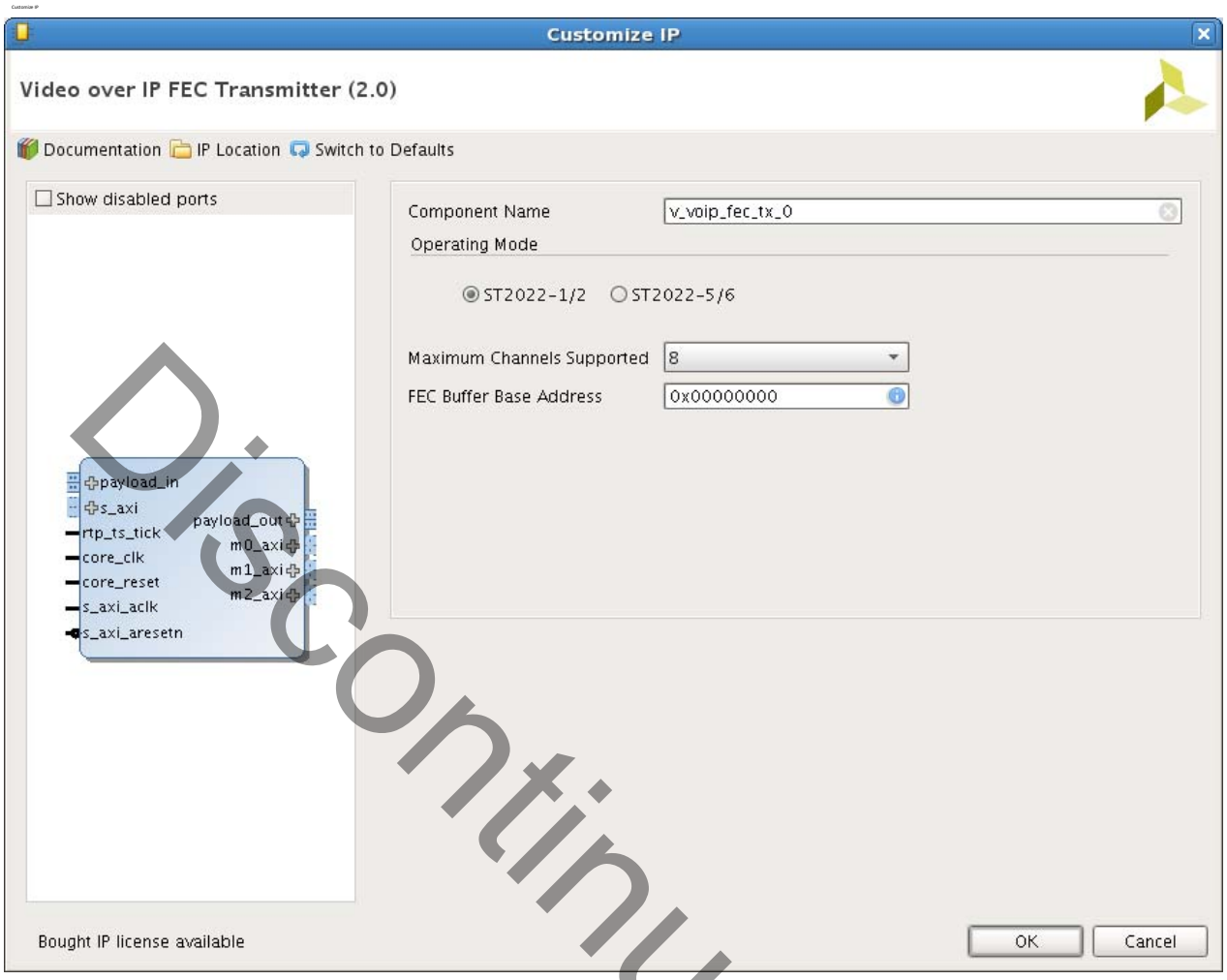


Figure 4-1: **Customize IP**

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

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

- **Component Name**: The base name of output files generated for the module. Names must begin with a letter and must be composed of characters a to z, 0 to 9 and "_". The name v_voip_fec_tx_v2_0 cannot be used as a component name.
- **Operating Mode**: Select the core operating mode. ST 2022-1/2, in which the core accepts SMPTE ST 2022-2 compliant RTP packet. ST 2022-5/6, in which the core accepts SMPTE ST 2022-6 compliant RTP packet.
- **Maximum Channels Supported**: Specifies the number of channels supported. The valid options for ST 2022-1/2 are 8, 16, 32, 64, 128, 256, 512, and for ST 2022-5/6 are 4 and 8.

- **FEC Buffer Base Address**: Specifies the base address of the FEC buffer in the external memory.

User Parameters

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

Table 4-1: **User Parameters**

GUI Parameter/Value	User Parameter/Value	Default Value	
		ST2022-1/2	ST2022-5/6
Maximum Channels Supported	C_CHANNELS	8	4
FEC Buffer Base Address	C_FEC_BASE_ADDR	0x00000000	0x00000000

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6]. The Vivado design tools generate the files necessary to build the core and place those files in the `<project>/<project>.srcs/sources_1/ip/<core>` directory.

Constraining the Core

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

Required Constraints

The constraints required for the core are clock frequency constraints described in Clocking in Chapter 3. Paths between two clock domains should be constrained with the `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 Placement

There is no specific clock placement requirement for this core.

Banking

There is no specific banking rule for this core.

Transceiver Placement

There is no specific Transceiver placement rule for this core.

I/O Standard and Placement

There is no specific I/O standard and placement rule for this core.

Simulation

This section contains information about simulating IP in the Vivado Design Suite. For comprehensive information about Vivado simulation components, as well as information about using supported third-party tools, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 9].

Synthesis and Implementation

This section contains information about synthesis and implementation in the Vivado Design Suite. For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 6].

Test Bench

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

As shown in Figure 5-1, the demonstration test bench is a simple System Verilog module which configures and tests the VoIP Forward Error Correction Transmitter core. The test bench consists of several modules like RTP packet generator for generating and driving RTP packets over transport stream, APIs and drivers for configuring the core, checker for integrity check of packet coming out of core. The test bench is supplied as part of the example simulation output product group.

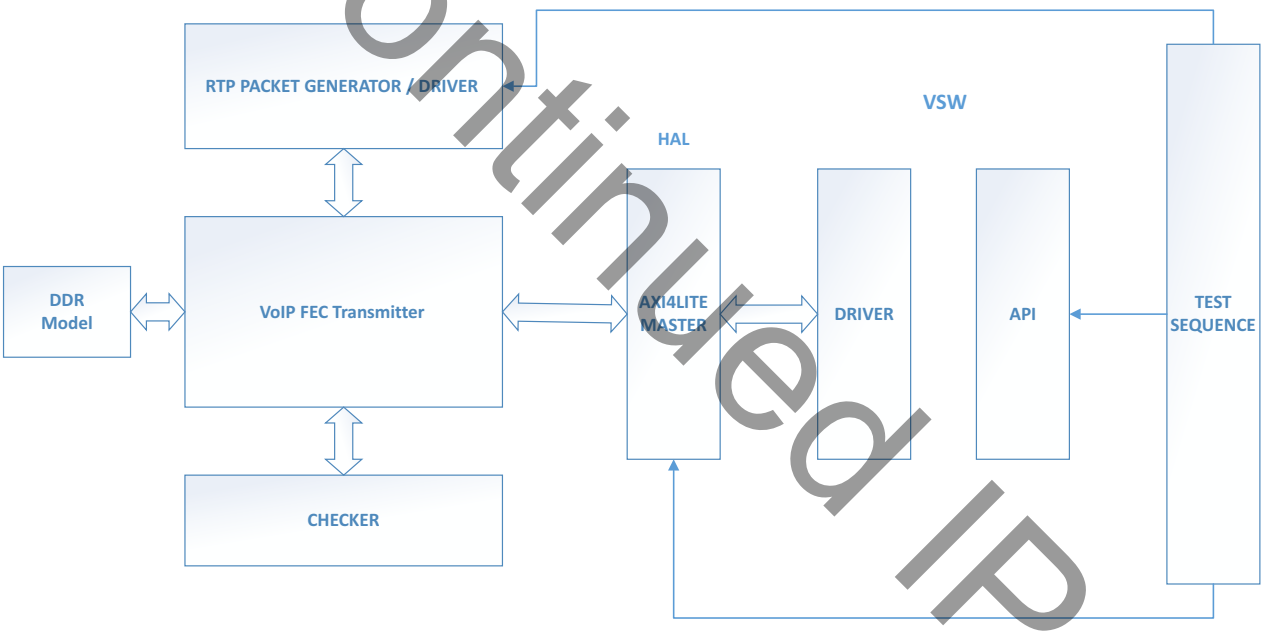


Figure 5-1: **Video over IP FEC Transmitter Test Bench**

The main components of Demonstration test bench are described below:

RTP Packet Generator/Driver: This module generates SMPTE ST 2022-2 and SMPTE ST 2022-6 compliant Real Time Protocol packets over Transport Stream and drives it to the Video over IP FEC Transmitter core across all the enabled channels.

Checker: Packet checker module receives the packet from the Video over IP FEC Transmitter core and does packet payload data integrity check, channel number mismatch check and packet size check.

HAL: Hardware Access Layer is the register configuration layer. This layer has register read and write process.

VSW: Virtual Software layer. This is a Verilog task file where all the core configuration is consolidated into tasks. This layer consists of Driver and API. They control the Core configuration and are driven to Core by HAL. This layer is controlled using test case.

DDR model: This is Dummy DDR model used to store the IP and FEC packets from core.

Appendix A

Verification, Compliance, and Interoperability

The Video over IP FEC Transmitter core has been validated using the Xilinx® Kintex®-7 FPGA Broadcast Connectivity Kit.

Migrating and Upgrading

This appendix contains information about migrating a design from the ISE® Design Suite 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.

Upgrading in the Vivado Design Suite

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

Changes from v1.0 to v2.0

Video over IP FEC Transmitter core version v2.0 had the following changes implemented, and may not compatible with previous core version, v1.0

Parameter Changes

Table B-1: **Parameter Changes**

Version		Note
V1.0	V2.0	
Component Name	Component Name	Unchanged
Maximum channels Supported	Maximum Channels Supported	Changed
FEC buffer base address	FEC Buffer Base Address	Changed

Port Changes

Table B-2: **Port Changes**

Version		Note
V1.0	V2.0	
soft_reset		Removed unused ports

Table B-2: **Port Changes *(Cont'd)***

Version		Note
interrupt		Removed unused ports
rtp_ts_clk	rtp_ts_tick	Changed

Other Changes

There are no other changes.

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 Video over IP FEC Transmitter, the Xilinx Support web page (Xilinx Support web page) 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 Video over IP FEC Transmitter. This guide, along with documentation related to all products that aid in the design process, can be found on the Xilinx Support web page or by using the Xilinx Documentation Navigator.

Download the Xilinx Documentation Navigator from the Downloads page. For more information about this tool and the features available, open the online help after installation.

Solution Centers

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

The Solution Center specific to the Video over IP FEC Transmitter core is listed below.

- Xilinx Ethernet IP Solution Center
- Xilinx MIG Solution Center

- Xilinx Solution Center for PCI Express

Answer Records

Answer Records include information about commonly encountered problems, helpful information on how to resolve these problems, and any known issues with a Xilinx product. Answer Records are created and maintained daily ensuring that users have access to the most accurate information available.

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

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

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

Master Answer Record for the Video over IP FEC Transmitter

AR: 54549

Technical Support

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

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

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

Debug Tools

There are many tools available to address Video over IP FEC Transmitter design issues. It is important to know which tools are useful for debugging various situations.

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also allows you to set trigger conditions to capture

application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx.

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

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

See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 11].

Reference Boards

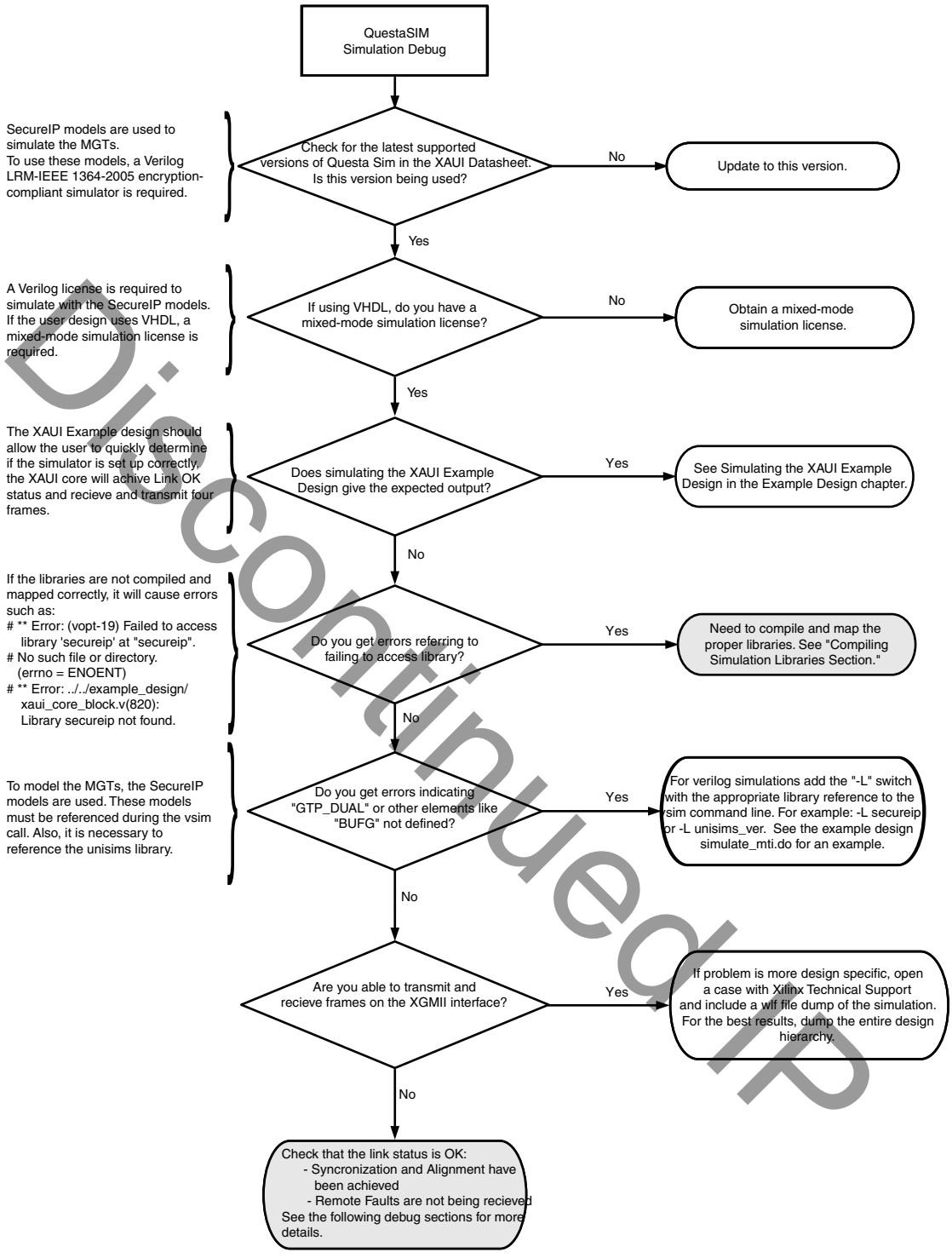
Various Xilinx development boards support the Video over IP FEC Transmitter. These boards can be used to prototype designs and establish that the core can communicate with the system.

- 7 series FPGA evaluation boards
  - KC705

Third-Party Tools

Simulation Debug

The simulation debug flow for Questa® SIM is illustrated in Figure C-1. A similar approach can be used with other simulators.



Hardware Debug

Hardware issues can range from link bring-up to problems seen after hours of testing. This section provides debug steps for common issues. The Vivado lab tools are a valuable resource to use in hardware debug. The signal names mentioned in the following individual sections can be probed using the Vivado lab tools for debugging the specific problems.

General Checks

Ensure that all the timing constraints for the core were properly incorporated from the example design and that all constraints were met during implementation.

- Does it work in post-place and route timing simulation? If problems are seen in hardware but not in timing simulation, this could indicate a PCB issue. Ensure that all clock sources are active and clean.
- If using MMCMs in the design, ensure that all MMCMs have obtained lock by monitoring the `locked` port.
- If your outputs go to 0, check your licensing.

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.

IP Core Debug

Crucial Core and Register Setting

1. When generating core in Vivado, ensure that the base address for FEC Buffer Base Address is set based on the recommended computations described in Memory Requirement in Chapter 3.
2. Ensure the core is not set on Error Configuration mode, which causes core to be in Bypass Mode.

   Error Configuration

   a. FEC D < 4

b. FEC L < 4 and FEC Mode is set to 2D

c. FEC L = 1; when Non-Block Align

3. If in_pkt_cnt (0x0024) register is incrementing, it indicates the core is receiving a valid UDP or RTP encapsulated ST 2022-2/6 compliant media packets.
4. If out_pkt_cnt (0x0028) register is incrementing, it indicates the core is transmitting a valid RTP encapsulated ST 2022-2/6 compliant media packet or RTP encapsulated ST 2022-1/5 compliant row FEC packet or RTP encapsulated ST 2022-1/5 compliant column FEC packet or UDP packets.

Additional Resources and Legal Notices and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support.

References

These documents provide supplemental material useful with this product guide:

1. SMPTE ST 2022-5,6 Video Over IP Product Page
2. *SMPTE ST 2022-1,2 Video Over IP* Product Page
3. *Modular Media over IP* Product Page
4. *Modular Media over IP Reference Design* (XAPP1272)
5. *Vivado® Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
6. *Vivado Design Suite User Guide: Designing with IP* (UG896)
7. *AXI Reference Guide* (UG1037)
8. *Vivado Design Suite User Guide: Getting Started* (UG910)
9. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
10. *ISE® to Vivado Design Suite Migration Guide* (UG911)
11. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
12. *Vivado Design Suite User Guide - Implementation* (UG904)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/05/2016	2.0	Added support for RFC3190 and RFC4175. Updated Xilinx automotive applications disclaimer.
04/06/2016	2.0	Updated for ST 2022-5/6 operation mode support.
11/18/2015	1.0	Added UltraScale+ support.
04/01/2015	1.0	Initial Xilinx release.

Please Read: Important Legal Notices