SMPTE 2022-1/2 Video over IP Receiver v2.0

LogiCORE IP Product Guide

Vivado Design Suite

PG181 October 5, 2016

XILINX
ALL PROGRAMMABLE™

Table of Contents

Introduction

The Xilinx LogiCORE™ IP SMPTE 2022-1/2 Video over IP Receiver core is used for broadcast applications that require bridging between constant bit rate MPEG-2 transport streams and 1 Gb/s IP networks. The module can recover IP packets lost due to network transmission errors and ensure integrity of transport streams. 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

- Up to 16 channels of CBR MPEG-2 transport streams in accordance with SMPTE 2022-2
- Per-channel forward error correction (FEC) in accordance with SMPTE 2022-1
- Level A and Level B FEC operations
- Block-aligned and non-block-aligned FEC operations support
- Supports Virtual Local Area Network (VLAN) filtering
- AXI4-Stream data interfaces
- AXI4-Lite control interface
- Configurable channel filtering based on any combinations of the following:
  - IP source address
  - IP destination address
  - User Datagram Protocol (UDP) source port
  - UDP destination port
  - Real-time Transport Protocol (RTP) Synchronization Source (SSRC) identifier
- VLAN tag value
- Seamless switching (SMPTE2022-7)
- RTP timestamp check bypass
- Include or remove FEC engine or secondary link during compile time

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family[1]	UltraScale+™ Families, UltraScale™ Architecture, Zynq-7000®, Virtex-7®, Kintex-7®, Artix-7®
Supported User Interfaces	AXI4-Lite, AXI4-Stream, AXI-4
Resources	See Table 2-1 through Table 2-3
Provided with Core	
Design Files	Encrypted HDL
Example Design	XAPP1194-kc705_smpte2022_12_4ch_rx
Test Bench	Verilog
Constraints File	XDC
Simulation Model	Encrypted RTL
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 at the Xilinx Support web page	

Notes:

1. For a complete list of supported devices, see Vivado IP catalog.
2. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.

Features (continued)

- Statistic indicators
  - Received packet
  - Reordered packet, Duplicated packet count
  - Recovered packet count
  - Valid packet count, Unrecoverable packet count
  - Out of range packet count
  - Packet interval measure
  - Buffer overflow flag
  - Seamless protect flag
  - Link differential measure

Overview

As broadcast and communications markets converge, broadcasters and telecommunication companies increasingly use IP networks for video stream transport. Xilinx devices bridge the broadcast and the communications industries by providing highly integrated real-time video interfaces that help broadcasters reduce costs and the time it takes to acquire, edit, and produce content.

Now that video can be delivered reliably over Ethernet, broadcasters can replace expensive mobile infrastructures that support outside live broadcasts, as well as enable remote production from existing fixed studios. This dramatically reduces both capital expenditure and operating expenses. As a result, using Ethernet to transmit multiple compressed media streams is a major customer requirement. The industry implements primarily the SMPTE 2022 set of standards to create an open and interoperable way to transmit video over Ethernet, ensure quality of service (QoS), and minimize packet loss.

As shown in Figure 1-1, SMPTE 2022-1/2 receiver core primarily targets distribution networks where multiple transport streams are carried over 1 Gb/s Ethernet networks. The core includes Forward Error Correction (FEC), which protects transport streams over IP networks. With FEC, the receiver adds systematically generated redundant data that allows the receiver to detect and correct a limited number of packet errors.

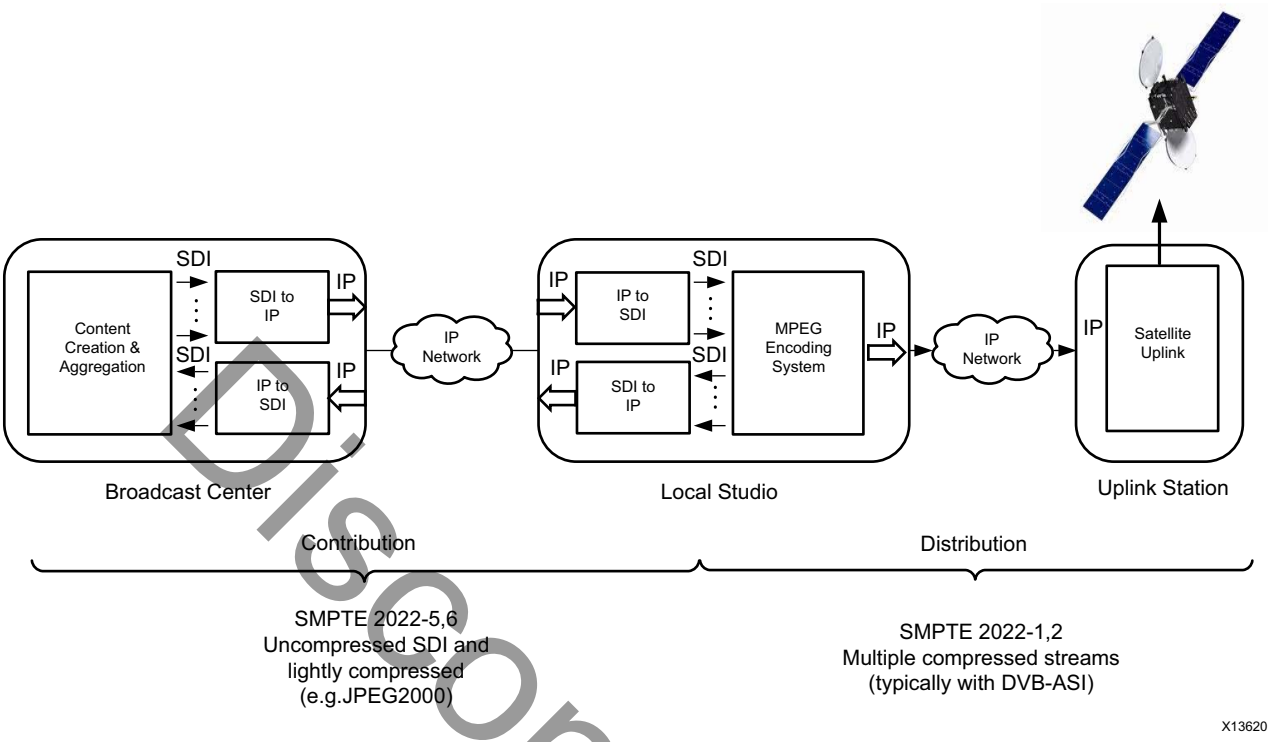

Figure 1-1: **SMPTE 2022-1/2 in Distribution Networks**

Video packets are lost for a variety of reasons, including thermal noise, storage system defects, and transmission noise introduced by the environment. FEC enables the receiver to correct these errors without using a reverse channel to request retransmission, which is not feasible in real-time systems because the latency is too great.

Feature Summary

The core de-capsulates Ethernet packets into transport streams and can recover IP packets lost because of network transmission errors, ensuring the data integrity of MPEG transport streams. The core operates seamlessly when receiving and filtering VLAN tagged Ethernet packets.

You configure and instantiate the core by using the Vivado® design tools. Core functionality is controlled through registers via an AXI4-Lite interface.

Applications

- Transport compressed constant bit rate MPEG-2 transport streams over IP networks.
- Support real-time audio/video applications in primary distribution.

Licensing and Ordering Information

This Vivado® Design Suite 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, see the SMPTE 2022-1/2 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.

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 tools:

- Vivado design 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

Figure 2-1 shows a block diagram of SMPTE 2022-1/2 Video over IP Receiver core.

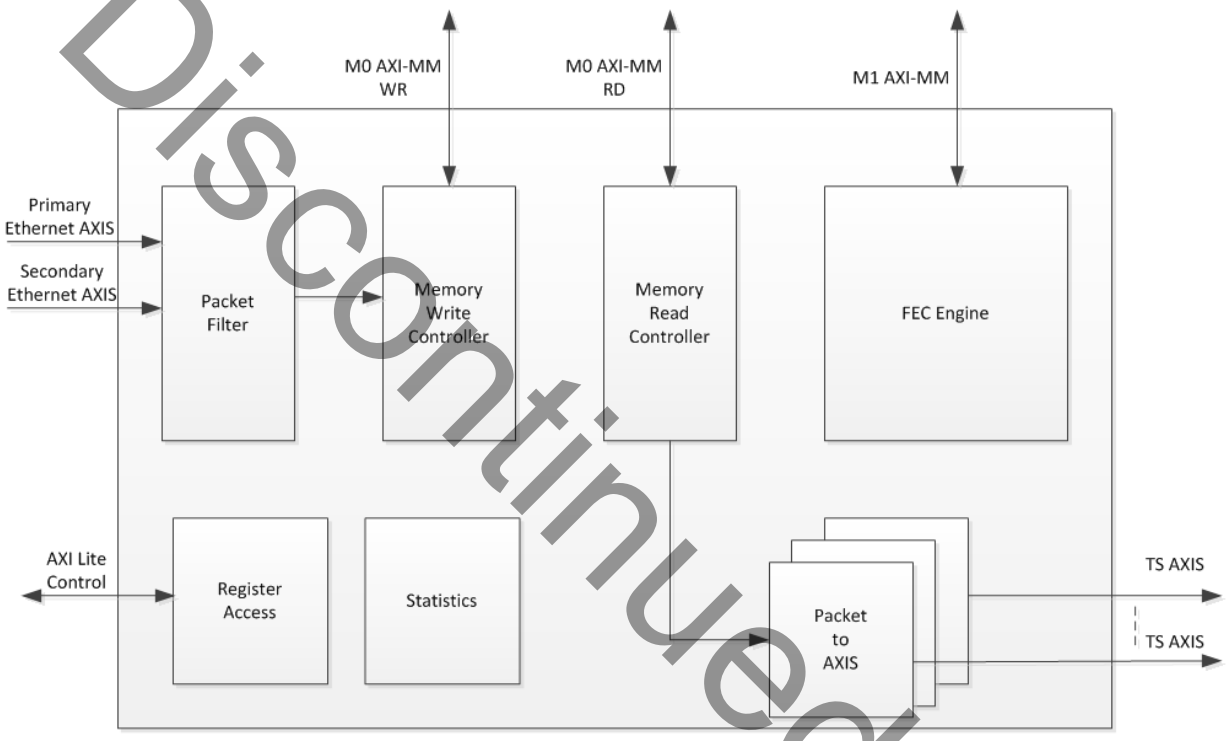

Figure 2-1: **Architecture Overview of SMPTE 2022-1/2 Video over IP Receiver Core**

The main functional blocks of the core are:

- Packet filter - Demultiplex the Ethernet packets into channel streams.
- Memory write controller - Puts Ethernet packets into the DDR memory buffer.
- Memory read controller - Retrieves Ethernet packets from the DDR memory buffer.
- Packet to AXIS - Convert the payload of the packets into AXI-Stream format
- FEC engine - Performs FEC recovery on packet loss in the DDR memory buffer.
- Register access - Register configuration and status read-back on the core.
- Statistics counters

Standards

The SMPTE 2022-1/2 Video over IP Receiver 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. The function of the core is compliant with SMPTE 2022-1/2 standard.

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. See Table 2-1 through Table 2-3 for device family–specific information.

Resource Utilization

Resources required for this core have been estimated for Zynq®-7000, Virtex-7®, Kintex-7®, and Artix®-7 devices. UltraScale™ results are expected to be similar to 7 series results. These values were generated using Xilinx Vivado® Design Suite. They are derived from post-synthesis reports, and might change during MAP and PAR.

Virtex-7 FPGAs

Table 2-1 provides approximate resource counts for the various core options on Virtex-7 FPGAs.

Table 2-1: **Resource Utilization for Virtex-7 FPGAs (xcv7vx690t Speed -1)**

CHANNEL	FEC INCLUDE	HITLESS INCLUDE	FFs	LUTs	Slices	LUT FF Pairs	36k BRAMs	18k BRAMs	DSP48E1s	Fmax (Mhz)
1	0	0	7007	3506	2489	6517	11	0	0	304
4	0	0	17293	9198	7166	16736	29	0	0	281
8	0	0	31010	16056	9717	27113	53	0	0	242
16	0	0	58460	29844	18412	51993	101	0	0	234
1	1	0	10928	5641	3277	9608	25	1	0	274
4	1	0	21211	11499	7407	19728	55	1	0	250
8	1	0	36012	18483	11613	32138	95	1	0	219
16	1	0	65629	33722	19761	57961	175	1	0	212

Table 2-1: **Resource Utilization for Virtex-7 FPGAs (xcv7vx690t Speed -1) *(Cont'd)***

CHANNEL	FEC INCLUDE	HITLESS INCLUDE	FFs	LUTs	Slices	LUT FF Pairs	36k BRAMs	18k BRAMs	DSP48E1s	Fmax (Mhz)
1	0	1	9571	4197	2841	8164	15	0	0	288
4	0	1	24994	10968	7381	21081	33	0	0	274
8	0	1	45489	19002	12907	37187	57	0	0	258
16	0	1	86521	35004	25338	71664	105	0	0	234
1	1	1	13895	6523	4102	11980	29	1	0	274
4	1	1	29650	13620	9545	26077	59	1	0	250
8	1	1	51973	22662	15422	43661	99	1	0	219
16	1	1	96567	41714	28936	81843	179	1	0	204

Kintex-7 FPGAs

Table 2-2 provides approximate resource counts for the various core options on Kintex-7 FPGA and Zynq-7000 Devices with Kintex Based Programmable Logic.

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

CHANNEL	FEC INCLUDE	HITLESS INCLUDE	FFs	LUTs	Slices	LUT FF Pairs	36k BRAMs	18k BRAMs	DSP48E1s	Fmax (Mhz)
1	0	0	6991	3493	2219	6247	11	0	0	288
4	0	0	17309	9210	5406	15458	29	0	0	266
8	0	0	31010	16059	10399	27865	53	0	0	242
16	0	0	58449	29409	17974	51182	101	0	0	219
1	1	0	10928	5638	3379	9739	25	1	0	274
4	1	0	21211	11494	6857	19053	55	1	0	258
8	1	0	36012	18480	11425	31903	95	1	0	219
16	1	0	65629	33722	20424	58369	175	1	0	196
1	0	1	9571	4194	2966	8309	15	0	0	296
4	0	1	24994	10964	6957	20830	33	0	0	258
8	0	1	45489	19000	14569	38308	57	0	0	234
16	0	1	86521	35006	25290	71142	105	0	0	226
1	1	1	13876	6515	4145	11869	29	1	0	266
4	1	1	29650	13620	9108	25671	59	1	0	242
8	1	1	51954	22407	15283	43267	99	1	0	219
16	1	1	96666	41480	28053	80508	179	1	0	156

Artix-7 FPGAs

Table 2-3 provides approximate resource counts for the various core options on Artix-7 FPGA and Zynq-7000 Devices with Artix Based Programmable Logic.

Table 2-3: **Resource Utilization for Artix-7 FPGAs (xc7a200t Speed -1)**

CHANNEL	FEC INCLUDE	HITLESS INCLUDE	FFs	LUTs	Slices	LUT FF Pairs	36k BRAMs	18k BRAMs	DSP48E1s	Fmax (Mhz)
1	0	0	7007	3504	2317	6341	11	0	0	204
4	0	0	17293	9195	5641	15772	29	0	0	180
8	0	0	31026	15926	10471	27906	53	0	0	172
16	0	0	58449	29413	17364	50687	101	0	0	156
1	1	0	10912	5662	3284	9533	25	1	0	204
4	1	0	21195	11482	7061	19270	55	1	0	180
8	1	0	36012	18484	11405	31763	95	1	0	148
16	1	0	65629	33681	18869	56810	175	1	0	140
1	0	1	9590	4207	3045	8332	15	0	0	204
4	0	1	24994	10967	7614	21142	33	0	0	196
8	0	1	45489	19012	14109	38217	57	0	0	180
16	0	1	86521	35977	23584	69319	105	0	0	164
1	1	1	13876	6548	4359	12017	29	1	0	196
4	1	1	29669	13635	8983	25417	59	1	0	172
8	1	1	51954	22361	15940	43779	99	1	0	156
16	1	1	96567	42730	26741	79352	179	1	0	148

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 derated 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 2022-1/2 Video over IP Receiver 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. The number of TX_AXIS interfaces depends on the number of channels configured through the GUI. SEC_ETH_AXIS interface is available when seamless switching is enabled. M1_AXI interface is enabled when the FEC engine is included.

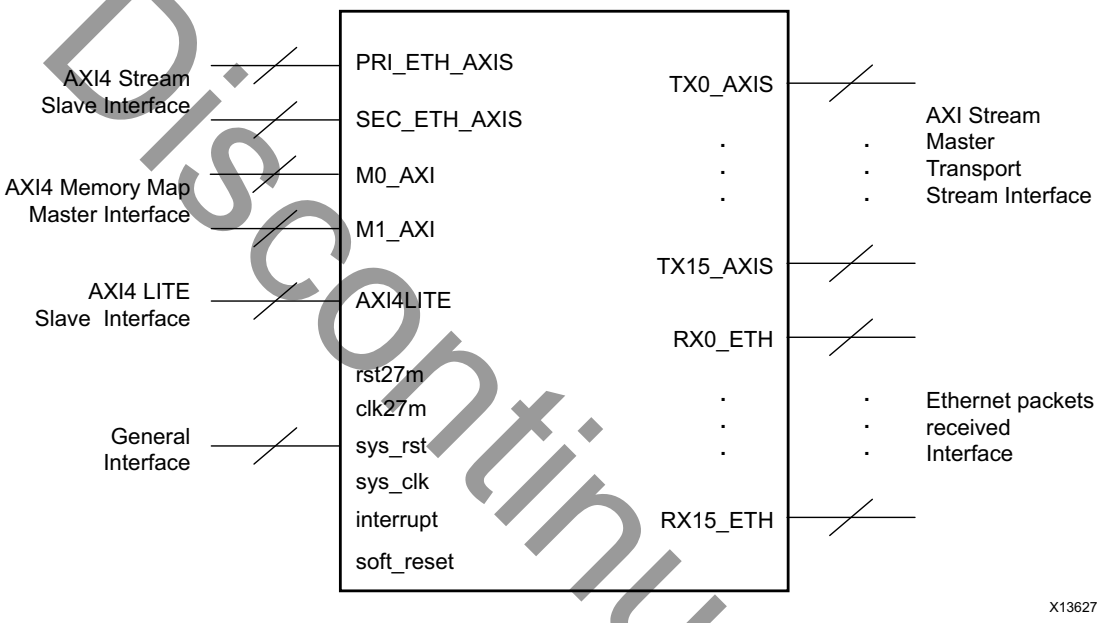

Figure 2-2: **SMPTE 2022-1/2 Video over IP Receiver Core Top Level Signaling Interface**

General Interface Signals

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

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

Signal Name	Direction	Width	Description
rst27m	In	1	27 MHz domain reset.
clk27m	In	1	27 MHz clock. Used for timekeeping.
sys_rst	In	1	System domain reset.
sys_clk	In	1	System clock.
Interrupt	Out	1	This signal is reserved for future use.
Soft_reset	Out	1	Core reset generated from specific control register bit.

AXI4 Memory Interface

The SMPTE 2022-1/2 Video over IP Receiver core uses an AXI4 interface to connect to the AXI4 interconnects. The AXI4 Interconnect provides access to external memory through the AXI 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	128	Write Data Channel Data.
m0_axi_wstrb	Out	16	Write Data Channel Data Byte Strobes.
m0_axi_wlast	Out	1	Write Data Channel Last Data Beat.
m0_axi_wvalid	Out	1	Write Data Channel Valid.
m0_axi_wready	In	1	Write Data Channel Ready.
m0_axi_bid	In	1	Write Response Channel Transaction ID.
m0_axi_bresp	In	2	Write Response Channel Response Code.
m0_axi_bvalid	In	1	Write Response Channel Valid.
m0_axis_bready	Out	1	Write Response Channel Ready.
m0_axi_arid	Out	1	Read Address Channel Transaction ID.
m0_axi_araddr	Out	32	Read Address Channel Address
m0_axi_arlen	Out	8	Read Address Channel Burst Length code.
m0_axi_arsize	Out	3	Read Address Channel Transfer Size code.
m0_axi_arburst	Out	2	Read Address Channel Burst Type.

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 Data Transaction ID.
m0_axi_rdata	In	128	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	128	Write Data Channel Data.
m1_axi_wstrb	Out	16	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	In	1	Read Address Channel Valid.
m1_axi_arready	In	1	Read Address Channel Ready.
m1_axi_rid	In	1	Read Data Channel Data Transaction ID.
m1_axi_rdata	In	128	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.

Explanation of the AXI_MM Port Function

Table 2-6: **AXI_MM Port**

Port	Description
m0 port write	Incoming packet processing
m0 port read	Outgoing packet processing
m1 port write	FEC recovery processing
m1 port read	m1 port read: FEC recovery processing

Ethernet AXI4 Stream Slave Interface

See the *LogiCORE Tri-Mode Ethernet MAC Product Guide* (PG051) [Ref 4] for more information.

Table 2-7: **AXI4 Stream Interface Signal**

Signal Name	Direction	Width	Description
pri_eth_rst	In	1	Active-High reset from TEMAC.
pri_eth_clk	In	1	Recovered clock from TEMAC.
pri_s_axis_tdata[7:0]	In	8	AXI4-Stream Data from TEMAC.
pri_s_axis_tvalid	In	1	AXI4-Stream Data Valid from TEMAC.
pri_s_axis_tlast	In	1	AXI4-Stream signal from TEMAC indicating an end of packet.
pri_s_axis_tuser	In	1	AXI4-Stream User Sideband Interface from TEMAC.
sec_eth_rst	In	1	Active-High reset from TEMAC.
sec_eth_clk	In	1	Recovered clock from TEMAC.
sec_s_axis_tdata[7:0]	In	8	AXI4-Stream Data from TEMAC.
sec_s_axis_tvalid	In	1	AXI4-Stream Data Valid from TEMAC.
sec_s_axis_tlast	In	1	AXI4-Stream signal from TEMAC indicating an end of packet.
sec_s_axis_tuser	In	1	AXI4-Stream User Sideband Interface from TEMAC.

AXI4-Lite Control Interface

The AXI4-Lite interface allows user to dynamically control parameters within the SMPTE 2022-1/2 Video over IP Receiver core. You can configure the core using an embedded ARM or soft system processor such as MicroBlaze.

You can control the core through the AXI4-Lite interface using read and write transactions to the SMPTE 2022-1/2 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, connecting via the AXI4-Lite interface to

an AXI4-Lite master. See the *LogiCORE IP AXI Interconnect Product Guide* (PG059) [Ref 3] for more information.

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

Signal Name	Direction	Width	Description
s_axi_clk	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_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
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.

AX4I-Stream Transport Interface.

Table 2-9: **Transport Stream AXI-Stream Interface Signals**

Signal Name	Direction	Width	Description
tx_axis_aresetn	Out	1	AXI4-Stream Active-Low reset.
tx_axis_aclk	In	1	AXI-4 Stream clock input.
tx_axis_tdata	Out	8	Transport stream Data out.
tx_axis_tvalid	Out	1	Transport stream Data Valid. A transfer takes place when both tx_axis_tvalid and tx_axis_tready are asserted.
tx_axis_tlast	Out	1	Indicates the boundary of a packet. Fixed at 0.
tx_axis_tuser	Out	1	User defined sideband information indicating synchronizing byte, 47h.
tx_axis_tready	In	1	Indicates that the slave can accept a transfer in the current cycle.

The transport stream data output behavior from the core transport stream interface is shown in Figure 2-3. The `tx_axis_tuser` signal is High when sending the synchronizing byte; `tx_axis_tvalid` is High when output data is valid; `tx_axis_tdata` is updated when `tx_axis_tready` is High; `tx_axis_tlast` is always Low, and there is no idle byte.

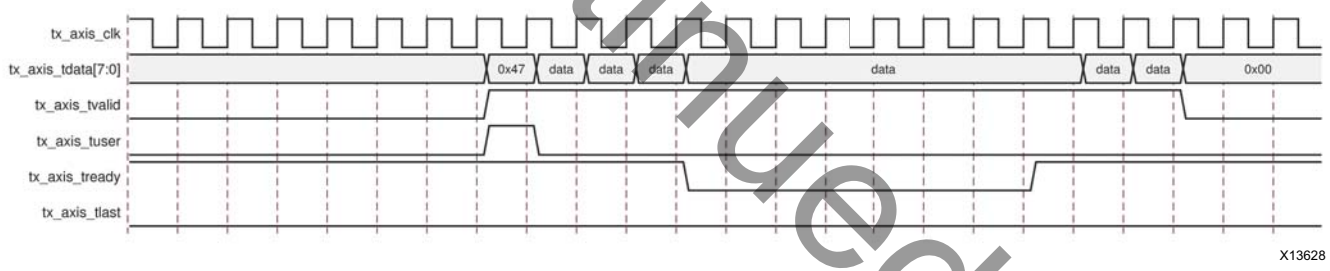

Figure 2-3: **SMPTE 2022-1/2 Video over IP Receiver Core Transport Stream AXI-Stream Interface**

Ethernet Packets Received Interface

Table 2-10: **Ethernet Packets Received Interface Signals**

Signal Name	Direction	Width	Description
rx_pri_rtp_pkt_recv	Out	1	Pulse indicating receiving of media packet from primary link. (Synchronous to pri_eth_clk)
rx_pri_rtp_seq_num	Out	16	Sequence number of media packet received from primary link. (Synchronous to pri_eth_clk)
rx_pri_rtp_ts	Out	32	Timestamp of the media packet received from primary link. (Synchronous to pri_eth_clk)
rx_sec_rtp_pkt_recv	Out	1	Pulse indicating receiving of media packets from secondary link. (Synchronous to sec_eth_clk)
rx_sec_rtp_seq_num	Out	16	Sequence number of media packet received from secondary link. (Synchronous to sec_eth_clk)
rx_sec_rtp_ts	Out	32	Timestamp of the media packet received from secondary link. (Synchronous to sec_eth_clk)
rx_rtp_pkt_buffered	Out	16	Amount of media packets buffered in the DDR for the channel. (Synchronous to sys_clk)
rx_rtp_pkt_transmit	Out	1	Pulse indicating consumption of media packet for TS output. (Synchronous to sys_clk)
rx_vid_lock	Out	1	Indication of channel locking to certain payload. (Synchronous to sys_clk)
rx_playout_ready	Out	1	Indication of channel ready for playing out the TS data. (Synchronous to sys_clk)

Register Space

The SMPTE 2022-1/2 Video over IP Receiver register space is partitioned into general and channel-specific registers.

Table 2-11: **AXI4-Lite Register Map**

Address (Hex) BASEADDR +	Register Name	Access Type	Default Value (HEX)	Register Description	
				Bit Range	Value
General Registers					

Table 2-11: **AXI4-Lite Register Map *(Cont'd)***

Address (Hex) BASEADDR +	Register Name	Access Type	Default Value (HEX)	Register Description	
				Bit Range	Value
0x000	control	R/W	0x00000000	**Control**	
				31:2	Reserved
				1	Channel Register Update. 0 – Host processor actively updating the channel registers 1 – Updating process completed
				0	Reserved
0x004	reset	R/W	0x00000000	**Reset**	
				31:1	Reserved
				0	1 – Reset the configuration registers and set soft_reset signal High
0x00C	channel_access	R/W	0x00000000	**Channel Access**	
				31	0 - primary 1 - secondary
				30:8	Reserved
				7:0	The channel number to access the channel space registers
0x020	sys_cfg	R		**System Configuration**	
				31	Hitless switching supported
				30	FEC recovery supported
				29:8	Reserved
				7:0	Number of channels supported
0x024	version	R	0x02000000	**Version**	
				31:24	Version major
				23:16	Version minor
				15:12	Version revision
				11:8	Patch ID
				7:0	Revision number

Table 2-11: **AXI4-Lite Register Map *(Cont'd)***

Address (Hex) BASEADDR +	Register Name	Access Type	Default Value (HEX)	Register Description	
				Bit Range	Value
0x028	network_path_differential	R/W	0x00000000	**Network Path Differential**	
				31:9	max acceptable delay between 2 streams in seamless mode, value based on 90kHz clock ticks
				8:0	Reserved
0x030	fec_pkt_processing_delay	R/W		**FEC Packet Processing Delay**	
				31:9	Set time delay for incoming FEC packets before processing for recovery for scenarios when packet arriving out of ordered or delayed packet. Value is counted based on 90 kHz clock tick.
0x034	fec_buf_base_addr	R/W	0x00000000	**FEC Buffer Base Address**	
				31:0	Base address on where the buffer begins in the DDR
0x038	fec_buf_pool_size	R/W	0x00000000	**FEC Buffer Pool Size**	
				31:0	No. of Bytes of memory space to cater for FEC buffer
0x03C	pri_recv_pkt_cnt	R		**Primary Received Packet Count**	
				31:0	Number of packets received in the primary stream
0x040	sec_recv_pkt_cnt	R		**Secondary Received Packet Count**	
				31:0	Number of packets received in the secondary stream
0x044	pri_err_pkt_cnt	R		**Primary Errored Packet Count**	
				31:0	Number of errored packets received in the primary stream

Table 2-11: **AXI4-Lite Register Map *(Cont'd)***

Address (Hex) BASEADDR +	Register Name	Access Type	Default Value (HEX)	Register Description	
				Bit Range	Value
0x048	sec_err_pkt_cnt	R		**Secondary Errored Packet Count**	
				31:0	Number of errored packets received in the secondary stream
0x04C	pri_discard_pkt_cnt	R		**Primary Discarded Packet Count**	
				31:0	Number of not matching pkts received in the primary stream
0x050	sec_discard_pkt_cnt	R		**Secondary Discarded Packet Count**	
				31:0	Number of not matching pkts received in the secondary stream
0x054	gen_stat_reset	R/W	0x00000000	**General Statistics Reset**	
				31:6	Reserved
				5	Reset sec_discard_pkts_cnt
				4	Reset pri_discard_pkts_cnt
				3	Reset sec_err_pkts_cnt
				2	Reset pri_err_pkts_cnt
				1	Reset sec_recv_pkts_cnt
				0	Reset pri_recv_pkts_cnt
Channel Registers					
0x084	ip_hdr_param	R		**IP Header Parameter**	
				31:16	Reserved
				15:8	type of service(TOS)
				7:0	time to live(TTL)

Table 2-11: **AXI4-Lite Register Map *(Cont'd)***

Address (Hex) BASEADDR +	Register Name	Access Type	Default Value (HEX)	Register Description	
				Bit Range	Value
0x088	match_vlan	R/W	0x00000000	**Match VLAN**	
				31	VLAN filtering 0 - Filter stream without VLAN, 1 - Filter stream with VLAN having tag info in bit 15:0
				30:16	Reserved
				15:0	16-bit VLAN tag info
0x08C	match_dest_ip_addr	R/W	0x00000000	**Match Destination IP Address**	
				31:0	32-bit destination IP address
0x09C	match_src_ip_addr	R/W	0x00000000	**Match Source IP Address**	
				31: 0	32-bit source IP address
0x0AC	match_src_port	R/W	0x00000000	**Match UDP Source Port**	
				31:16	Reserved
				15:0	16-bit UDP source port address
0x0B0	match_dest_port	R/W	0x00000000	**Match UDP Destination Port**	
				31:16	Reserved
				15:0	16-bit UDP destination port address
0x0B4	match_sel	R/W	0x00000000	**Matching Selection**	
				31:6	Reserved
				5	To match SSRC
				4	To match UDP dest port
				3	To match UDP src port
				2	To match Destination IP
				1	To match Source IP
				0	To match VLAN

Table 2-11: **AXI4-Lite Register Map *(Cont'd)***

Address (Hex) BASEADDR +	Register Name	Access Type	Default Value (HEX)	Register Description	
				Bit Range	Value
0x0B8	link_reordered_ pkt_cnt	R		**Link Reordered Packet Count**	
				31:0	Number of reordered packets
0x0BC	link_stat_reset	R/W	0x00000000	**Link Statistics Reset**	
				31:3	Reserved
				2	Reset link_valid_fec_pkt_cnt
				1	Reset link_valid_media_pkt_cnt
				0	Reset link reordered_pkt_cnt
0x0C0	link_valid_media_pkt_ cnt	R		**Link Valid Media Packet Count**	
				31:0	Number of valid media packets received in the link per channel
0x0C4	link_valid_fec_pkt_cnt	R	31:0	**Link Valid Media Packet Count**	
					Number of valid FEC packets received in the channel per link
0x100	chan_en	R/W	0x00000000	**Channel Enable**	
				31:2	Reserved
				1	RTP timestamp bypass 1 - operate on RTP stream with no timestamp 0 - operate on RTP stream with timestamp
				0	Channel enable 0-disablechannel. 1 - enable channel

Table 2-11: **AXI4-Lite Register Map *(Cont'd)***

Address (Hex) BASEADDR +	Register Name	Access Type	Default Value (HEX)	Register Description	
				Bit Range	Value
0x10C	chan_stat_reset	R/W		**Channel Statistics Reset**	
				31:6	Reserved
				5	Reset oor_ pkt_cnt
				4	Reset unrec_pkt_cnt
				3	Reset media_buffer_ov
				2	Reset dup_pkt_cnt
				1	Reset corr_ pkt_cnt
				0	Reset chan_valid_media_pkt_cnt
0x110	match_ssrc	R/W	0x00000000	**Match SSRC**	
				31:0	32-bit SSRC value used to match between primary and secondary links to the channel
0x11C	playout_delay	R/W	0x00000000	**Playout delay**	
				31:9	Set wait time to TS playout upon incoming stream packet size lock. Value based on 90kHz clock ticks
				8:0	Reserved
0x120	ts_status	R	0x00000000	**Transport Stream Status**	
				31:5	Reserved
				4:2	TS Packets per IP (1..7)
				1	TS pkt size: 0-188bytes; 1-204 bytes
				0	Packet size locked indicator

Table 2-11: **AXI4-Lite Register Map *(Cont'd)***

Address (Hex) BASEADDR +	Register Name	Access Type	Default Value (HEX)	Register Description	
				Bit Range	Value
0x124	fec_param	R	0x00000000	**FEC parameter**	
				31:22	Reserved
				21	FEC protect level. 1 - level B. 0 - level A.
				20	1 - FEC parameters locked
				19:10	10-bit D value from the received header
				9:0	10-bit L value from the received header
0x128	seamless_protect	R		**Seamless Protect**	
				31	Seamless status 0- not protected. 1 - protected
				30:0	RTP timestamp difference between incoming primary and secondary stream packets
0x12C	media_buf_base_addr	R/W	0x00000000	**Media Buffer Base Address**	
				31:0	Base address on where the buffer begins in the DDR
0x130	media_pkt_buf_size	R/W	0x00000000	**Media Packet Buffer Size**	
				31:16	Reserved
				15:0	The number of RTP packets to hold in the DDR for the channel
0x134	chan_valid_media_pkt_cnt	R		**Channel Valid Media Packet Count**	
				31:0	Number of valid media packets received in the channel
0x138	rec_pkt_cnt	R		**Recovered Packet Count**	
				31:0	Number of FEC recovered packets

Table 2-11: **AXI4-Lite Register Map *(Cont'd)***

Address (Hex) BASEADDR +	Register Name	Access Type	Default Value (HEX)	Register Description	
				Bit Range	Value
0x13C	dup_pkt_cnt	R		**Duplicated Packet Count**	
				31:0	Number of duplicated packets in the channel
0x144	pkt_interval	R		**Packet interval**	
				31:0	Timestamp difference between 2 media packets' consecutive sequence numbers in the merge stream
0x154	media_buffer_ov	R		**Media Buffer Overflow**	
				31:1	Reserved
				0	1 - indicate that media buffer overflowed
0x158	unrec_pkt_cnt	R		**Unrecoverable Packet Count**	
				31:0	Number of unrecoverable packets
0x160	oor_pkt_cnt	R		**Out-Of-Range Packet Count**	
				31:0	Number of out-of-range packets

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, the configuration registers are held at reset state. At the same time, the `soft_reset` signal at the core interface is held High.

CHANNEL_ACCESS (0x00C) Register

Set the channel to access. All the primary link and secondary link channels share the same set of register address in the channel space. To access secondary link channels, set bit 31 to 1. Only 0x084 - 0x0C0 registers are available for secondary link.

SYS_CFG (0x020) Register

System configuration of the core.

- Bit 31 High indicates seamless switching support.
- Bit 30 High indicates FEC engine is included.
- Bit 7-0 gives the number of channels available to use.

VERSION (0x024) 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.

NETWORK_PATH_DIFFERENTIAL (0x028) Register

Set the maximum delay between primary and secondary link for the core to operate in seamless switching mode. The value is based on a 90 kHz clock tick in Bit 31-9.

FEC_PKT_PROCESSING_DELAY (0X030) Register

Set time delay for incoming FEC packets before processing for recovery for scenarios in which packets arriving out of order or delayed packets. Value is count based on 90 kHz clock tick.

FEC_BUF_BASE_ADDR (0x034) Register

This sets the base address of the memory allocated in the DDR to store the FEC packets for recovery.

FEC_BUF_POOL_SIZE (0x038) Register

This register allocates the memory buffer size in the DDR for FEC packets storage. The value is in terms of bytes.

PRI_RECV_PKT_CNT (0x03C) Register

Primary received packet counter increments when a packet is filtered into the channels in the primary link. Register 0x054, bit 0 resets it.

SEC_RECV_PKT_CNT (0x040) Register

Secondary received packet counter increments when a packet is filtered into the channels in the secondary link. Register 0x054, bit 1 resets it.

PRI_ERR_PKT_CNT (0x044) Register

Primary error packet counter increments when a packet is identified as bad frame from the MAC core in the primary link. Register 0x054, bit 2 resets it.

SEC_ERR_PKT_CNT (0x048) Register

Secondary error packet counter increments when a packet is identified as bad frame from the MAC core in the secondary link. Register 0x054, bit 3 resets it.

PRI_DISCARD_PKT_CNT (0x04C) Register

Primary discard packet counter increments when a packet is not accepted for any of the channels in the primary link. Register 0x054, bit 4 resets it.

SEC_DISCARD_PKT_CNT (0x050) Register

Secondary discard packet counter increments when a packet is not accepted for any of the channels in the secondary link. Register 0x054, bit 5 resets it.

GEN_STAT_RESET (0x054) Register

- A High to Bit 5 resets secondary discarded packet counter (register 0x050).
- A High to Bit 4 resets primary discarded packet counter (register 0x04C).
- A High to Bit 3 resets secondary error packet counter (register 0x048).
- A High to Bit 2 resets primary error packet counter (register 0x044).
- A High to Bit 1 resets secondary received packet counter (register 0x040).
- A High to Bit 0 resets primary received packet counter (register 0x03C).

IP_HDR_PARAM (0x084) Register

Read-only status on the IP header fields.

- Bit 15-8 is Type of service (TOS).
- Bit 7-0 is Time to live (TTL).

MATCH_VLAN (0x088) Register

This parameter is used in filtering the packets for the channel. Configure Bit 15-0 for VLAN tag matching. Set Bit 31 for valid VLAN tag.

If Bit 31 is High, packets with VLAN tag and matching information are filtered into the channel.

If Bit 31 is Low, packets without VLAN tag is filtered into the channel.

Table 2-12: **Usage of Match VLAN (0x88) with Match Select (0xB4)**

Match Select (0xB4) Bit '0' To Match VLAN	Match VLAN (0x88) Bit '31' VLAN Filtering	Incoming Packet with VLAN	Incoming Packet without VLAN
0	0	Not applicable	
	1		
1	0	Not Match	Match
	1	Match if VLAN TAG ID = Bit [15:0] of Match VLAN	Not Match

MATCH_DEST_IP_ADDR (0x08C) Register

This parameter is used in filtering the packets for the channel. Configure Bit 31-0 for Destination IP address matching.

MATCH_SRC_IP_ADDR (0x09C) Register

This parameter is used in filtering the packets for the channel. Configure Bit 31-0 for Source IP address matching.

MATCH_SRC_PORT (0x0AC) Register

This parameter is used in filtering the packets for the channel. Configure Bit 15-0 for UDP Source Port matching.

MATCH_DEST_PORT (0x0B0) Register

This parameter is used in filtering the packets for the channel. Configure Bit 15-0 for UDP Destination Port matching.

MATCH_SEL (0x0B4) Register

This register sets the parameters to match the incoming packet and filter it to the channel.

- Bit 5 matches the SSRC field in the RTP header of the packet to the configured register 0x110.
- Bit 4 matches the Destination Port field in the UDP header of the packet to the configured register 0x0B0.
- Bit 3 matches the Source Port field in the UDP header of the packet to the configured register 0x0AC.
- Bit 2 matches the Destination address field in the IP header of the packet to the configured register 0x08C.
- Bit 1 matches the Source address field in the IP header of the packet to the configured register 0x09C.
- Bit 0 matches the IEEE 802.1Q tag in the Ethernet frame to the configured register 0x088.

Note that each link media packet can only be filtered into 1 particular channel.

LINK_REORDERED_PKT_CNT (0x0B8) Register

This counter tracks the number of incoming media packets that are reordered in the link channel (primary or secondary). A packet is considered reorder when its sequence number is less than the previous packet. Register 0x0BC, bit 0 resets the counter.

LINK_STAT_RESET (0x0BC) Register

- A High Bit 2 resets link valid FEC packet counter (register 0x0C4).
- A High to Bit 1 resets link valid media packet counter (register 0x0C0).
- A High to Bit 0 resets link reordered packet counter (register 0x0B8).

LINK_VALID_MEDIA_PKT_CNT (0x0C0) Register

This counter tracks the number of incoming media packets that match to the channel in the link (primary or secondary). Register 0x0BC, bit 1 resets the counter.

LINK_VALID_FEC_PKT_CNT (0X0C4) Register

This counter tracks the number of incoming FEC packets that match the channel in the link (primary or secondary). Register 0x0BC, bit 2 resets the counter.

CHANNEL_ENABLE (0x100) Register

- Set Bit 0 to enable the channel operation.
- Set Bit 1 to bypass media packet RTP timestamp check for the channel. Out-of-range counter is not active. Out-of-range packets are not discarded.

CHAN_STAT_RESET (0x10C) Register

- A High to Bit 5 resets out-of-range counter (register 0x160).
- A High to Bit 4 resets unrecoverable packet counter (register 0x158).
- A High to Bit 3 resets media buffer overflow (register 0x154).
- A High to Bit 2 resets duplicated packet counter (register 0x13C).
- A High to Bit 1 resets recovered packet counter (register 0x138).
- A High to Bit 0 resets channel valid media packet counter (register 0x134).

MATCH_SSRC (0x110) Register

This parameter is used in filtering the packets for the channel. Configure Bit 31-0 for RTP Synchronization Source identifier matching.

PLAYOUT_DELAY (0x11C) Register

Sets the wait time before the buffered Transport Stream data is ready for play out after packet size lock (register 0x120, bit 0). The value is based on 90kHz clock tick in Bit 31-9.

TS_STATUS (0x120) Register

Read only status on the media packet size detected by the channel.

Bit 0 is High after detection of 33 consecutive media packets with the same size for the channel. Change of packet size thereafter resets this bit and restart the whole process of detection.

Note: If packet lock is Low, all incoming packets are dropped.

When Bit 0 packet lock is High:

- Bit 4-2 shows the number of Transport Stream packets in the media packet.
- Bit 1 shows the Transport Stream packet size. Bit 1 is Low for 188 bytes and High for 204 bytes.

FEC_PARAM (0x124) Register

Read-only status on the FEC parameters detected by the channel.

Bit 20 is High upon first detection of FEC packet for the channel. Detection does not start before register 0x120, bit 0 is High.

When Bit 20 is High:

- Bit 9-0 shows the L value of FEC matrix.
- Bit 19-10 shows the D value of FEC matrix.
- Bit 21 shows the FEC protection level. Bit 21 is Low for Level A stream and High for Level B.

SEAMLESS_PROTECT (0x128) Register

Bit 30-0 samples the RTP timestamp difference between incoming packets from the primary and secondary links of a channel.

Bit 31 indicates whether the channel is under seamless protection. The channel is considered protected when the RTP timestamps of the media packets from both primary and secondary links fall within the range defined by `NETWORK_PATH_DIFFERRENTIAL` and `PLAYOUT_DELAY`.

MEDIA_PKT_BASE_ADDR (0x12C) Register

This register sets the base address of the memory buffer allocated in the DDR to store the media packets for FEC recovery and play out.

MEDIA_PKT_BUF_SIZE (0x130) Register

This register sets the maximum number of media packets to be stored in the DDR for the channel. It has a limitation of 16 bits (Bit 15-0) and has to be in written in the value of (2^n-1).

CHAN_VALID_MEDIA_PKT_CNT (0x134) Register

This counter increases when a media packet gets written into the memory buffer. It is per channel and register 0x10C, bit 0 resets it.

REC_PKT_CNT (0x138) Register

Recovered packet counter increases when a media packet is being recovered by the FEC engine. This counter is per channel and register 0x10C, bit 1 resets it.

DUP_PKT_CNT (0x13C) Register

Duplicated packet counter increases when the combined primary and secondary link of the channel has received a media packet with a sequence number that is already in the memory buffer. This incoming packet is discarded and not processed further. The counter is per channel and register 0x10C, bit 2 resets it.

PKT_INTERVAL (0x144) Register

The difference in media packets' timestamps between 2 consecutive sequence numbers in the media packets stream. The value is in terms of 90kHz clock tick.

MEDIA_BUF_OV (0x154) Register

Bit 0 of this register flags High when incoming packets fill up the memory buffer faster than the outgoing packets can clear. This status is per channel and register 0x10C, bit 3 resets it.

UNREC_PKT_CNT (0x158) Register

Unrecoverable packet counter increases when the outgoing packet is not available. There is no previous incoming packet or the FEC engine is not able to recover the missing packet from the matrix. This counter is per channel and register 0x10C, bit 4 resets it.

OOR_PKT_CNT (0x160) Register

Out-of-range packet counter increases when the difference between the incoming and outgoing packets' RTP timestamp is greater than the value of (NETWORK_PATH_DIFFERRENTIAL + PLAYOUT_DELAY). The incoming packet is discarded and not processed further. This counter is per channel and register 0x10C, bit 5 resets it.

IMPORTANT: *Note that massive out-of-range packets may cause the core to stop working.*

Register Configuration

For general registers, normal address read and write access is applied.

For channel registers, perform the following steps to update the registers:

1. Set the channel number at channel_access (0x00C) register.
2. Select either the Primary or Secondary link to be configured using the most significant bit of the channel_access (0x00C) register.
3. Configure the channel-specific register.

- Register Space 0x100-160 (Configure during Primary Link Configuration)
- Register Space 0x84-0xC4 (Configure during Primary and Secondary Link Configuration)

4. Pulse bit 1 of control (0x000) register to commit the channel registers change.
5. Repeat step 1 through step 4 for another channel or registers or link configuration. See Figure 2-4.

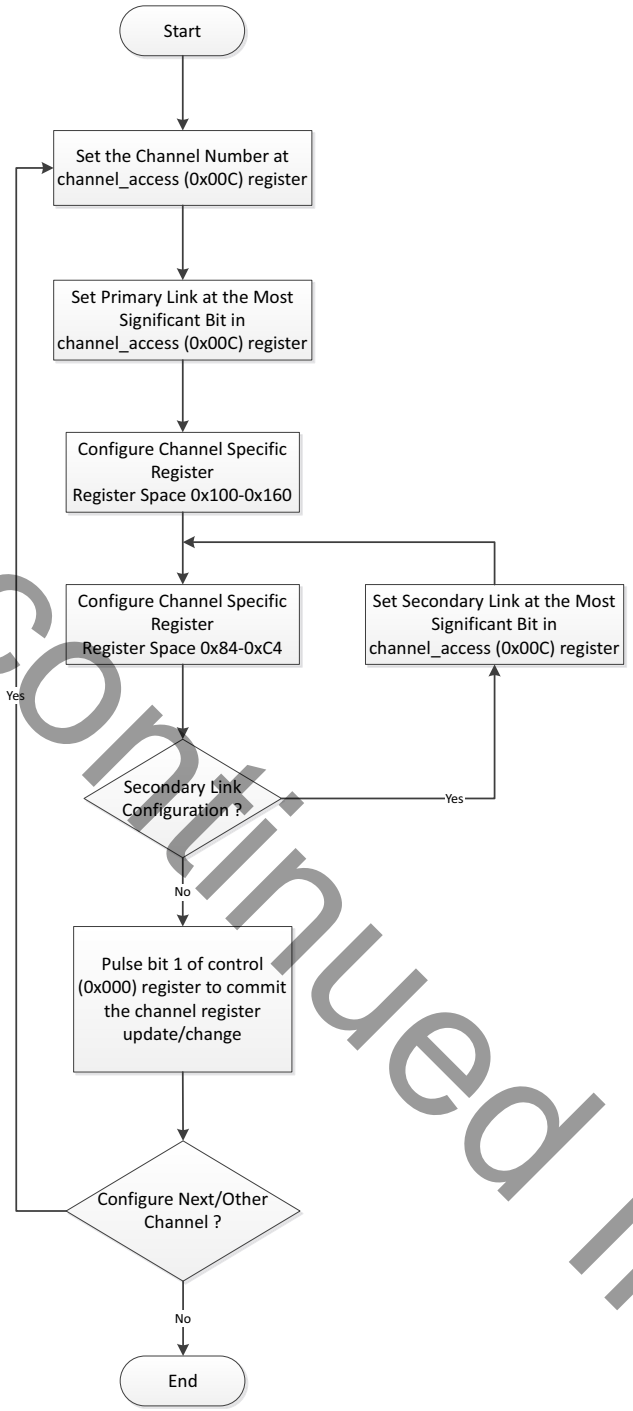

Figure 2-4: **Register Configuration**

Designing with the Core

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

General Design Guidelines

The SMPTE 2022-1/2 Video over IP Receiver core is designed for broadcast applications that require bridging between the transport stream and 1 Gb/s Ethernet. The core accepts Ethernet packets encapsulated in accordance with SMPTE 2022-1/2 and extracts the transport stream packets. The packets are sent out via an AXI stream interface to multiple channels of ASI videos with DVB-ASI cores. The core receives Ethernet packets through an AXI4-Stream interface from 1 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 the AXI4-Lite interface.

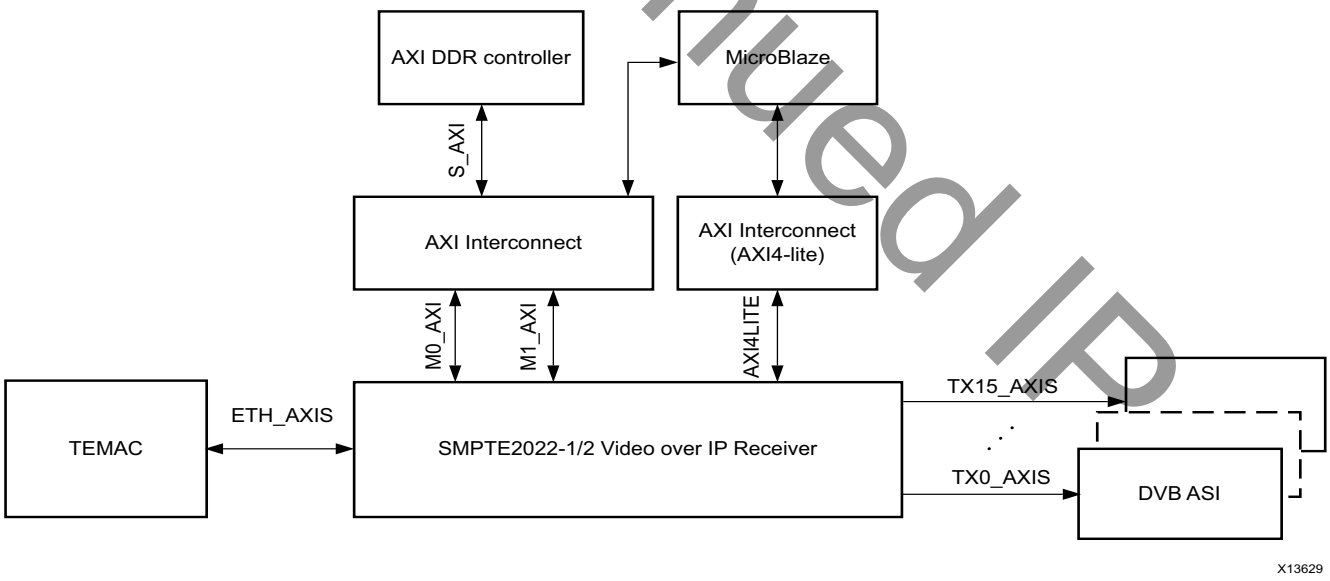

Figure 3-1: **SMPTE 2022-1/2 Video over IP Receiver System Built with Other Xilinx IP**

Note: In the SMPTE 2022-1/2 Video over IP Receiver core, you can include the FEC engine or enable seamless switching. FEC ensures the quality of compressed video by allowing the receiver to recover IP packets lost to network transmission errors. However, FEC increases the resource count in the device as well as the usage of external memory. Enabling seamless switching adds a redundancy

protection link for packets lost to network transmission errors, which also increases device resource count.

Clocking

The SMPTE 2022-1/2 Video over IP Receiver core has six main clock domains:

- Transport stream clock domain, tx_axis_aclk (recommended 148.5Mhz)
- System clock domain, sys_clk (refer to Fmax data)
- Primary link Ethernet clock domain, pri_eth_clk (125Mhz)
- Secondary link Ethernet clock domain, sec_eth_clk (125mhz)
- 27 MHz clock domain (clk27m)
- AXI4-Lite clock domain, s_axi_aclk (recommended 100Mhz)

Resets

The SMPTE 2022-1/2 Video over IP Receiver core has four main resets:

- Primary Ethernet link reset, pri_eth_rst
- Secondary Ethernet link reset, sec_eth_rst
- System domain reset, sys_rst
- 27Mhz domain reset, rst27m
- AXI4-Lite domain reset, s_axi_aresetn

The resets must be synchronous to their individual clock domains. A minimum of 16 clock cycles is recommended for the reset assertion. The pri_eth_rst and sec_eth_rst resets must be deasserted last.

Memory Requirements

The amount of DDR memory required by the SMPTE 2022-1/2 Video over IP Receiver core is determined by the number transport stream packets in an IP packet and the transport stream packet size for each channel.

Base Address and Play out Delay Register Setting

Follow the computation below to get the configuration values to obtain the channel packet buffer size and play out delay (**in bold**).

Note: all computation below are per-channel basis.

$$Video\ Packet_{rate} = \frac{TS_{rate}}{TS_{perIP} * TS_{size} * 8}$$

First compute packets buffered for Network Path Differential;

$*Network\ Path\ Differential_{in\ seconds}$ is as maximum accepted delay between 2 streams in Seamless Switching Mode or maximum accepted jitter in a single link.

$$Packet\ Buffered_{Network\ Path\ Differential} = (Network\ Path\ Differential_{seconds} + Network\ Jitter_{seconds}) * Video\ Packet_{rate}$$

Note: $Network\ Jitter_{seconds}$ is the maximum jitter in a single link network.

Then compute packet to buffered for FEC correction;

$$Packet\ Buffered_{FEC\ Correction} = (FEC_L * FEC_D * 2) + Packet\ Margin$$

Packet margin is set 32 as to add time for FEC recovery process.

The summed packet buffered is computed as;

The summed packet buffered is computed as;

$$Summed\ Packet\ Buffered = (Packet\ Buffered_{Network\ Path\ Differential} * 2) + Packet\ Buffered_{FEC\ Correction}$$

1. To calculate Media Packet Buffer Size,

*the maximum Media Packet Buffer Size is 65535 (0xFFFF)

$$Media\ Packet\ Buffer\ Size = 2^{[log_2(Summed\ Packet\ Buffered_{Ceiling})]} - 1$$

2. To calculate the play out delay based on the obtained information.

$$Playout\ Delay\ (seconds) = \frac{(Packet\ Buffered_{Network\ Path\ Differential} + Packet\ Buffered_{FEC\ Recovery} + Media\ Packets_{FEC\ Processing\ Delay})}{Video\ Packet_{rate}}$$

Notes:

1. $Media\ Packets_{FEC\ Processing\ Delay}$ is number of ST2022-2 Media Packets is being hold during FEC Processing Delay.
2. $Packet\ Buffered_{FEC\ Recovery}$ is minimum ST2022-2 Media Packets required for FEC Recovery.

To convert Playout Delay to 90 kHz clock ticks,

$$\textbf{\textit{Playout Delay (ticks)}} = \textit{Playout Delay(seconds)} * 90000\ Hz$$

Follow the computation in Table 3-1 to determine the FEC Buffer Base Address in the general space and Media Buffer Base Address per channel in the channel space (in **bold**).

Table 3-1: **Look up Table for Size Allocated per Media (ST2022-2) Packet in the Memory**

Media (TS Size = 188 Bytes)		Media (TS Size = 204 Bytes)	
TS packet per IP	**Size allocated per Packet (Bytes)**	**TS packet per IP**	**Size allocated per Packet (Bytes)**
1	256	1	272
2	432	2	464
3	624	3	672
4	816	4	896
5	1024	5	1088
6	1184	6	1280
7	1376	7	1536

Note: In Table 3-1, Media = SMPTE ST2022-2 packet.

Based on Table 3-1, compute the size allocated in the memory for each (ST2022-2) media channel:

$$\textit{Size per Channel}_{media} = \textit{Size allocated per Packet (Bytes)}_{media} * \textit{Media Packet Buffer Size}$$

3. Media Buffer Base Address for each channel can be set consecutively by,

$$\textbf{\textit{Media Buffer Base Address}}_{\textbf{Channel n}} = \textit{Media Buffer Base Address}_{Channel\ n-1} + \textit{Size per Channel}_{media\ n-1}$$

4. FEC Buffer Base Address can be set after the media buffer allocation by,

$$\textbf{\textit{FEC Buffer Base Address}} \geq \textit{Media Buffer Base Address}_{N\ Channel} + \textit{Size per Channel}_{media\ (N\ Channel)}$$

$$\textbf{\textit{FEC Buffer Base Address}} \leq \textit{Media Buffer Base Address}_{N\ Channel} - \textit{FEC Buffer Pool Size}$$

Note: The N channel is the last channel configuration for the core.

Note: The equation above ensures that the FEC Buffer Base Address does not overlap with the Media Buffer Base Address.

$$FEC\ Packet\ Buffered_{channel} = ((FEC\ L + FEC\ D) * 2) + (Packet\ Margin) + (FEC\ Packets_{FEC\ Processing\ Delay})$$

Notes:

1. FEC $Packets_{FEC\ Processing\ Delay}$ is number of ST2022-1 FEC Packets required to be hold before FEC recovery.
2. [(FEC L+FEC D)*2] is minimum ST2022-1 FEC Packets required for FEC recovery.
3. Packet Margin is set to 16 ST2022-1 FEC Packets.

Table 3-2: **Look up Table for Size Allocated per FEC (ST2022-1) Packet in the Memory**

FEC (TS Size = 188 Bytes)		FEC (TS Size = 204 Bytes)	
TS packet per IP	**Size allocated per Packet (Bytes)**	**TS packet per IP**	**Size allocated per Packet (Bytes)**
1	272	1	288
2	448	2	480
3	640	3	688
4	832	4	896
5	1024	5	1104
6	1200	6	1296
7	1408	7	1536

Note: In Table 3-2, FEC = SMPTE2022-1 packet.

Based on Table 3-2, compute the size allocated in the memory for FEC Pool Size per channel:

$$FEC\ Pool\ Size = Size\ allocated\ per\ Packet\ (Bytes)_{FEC} * FEC\ Packet\ Buffered_{Channel}$$

5. Summing the each pool size,

$$\boldsymbol{FEC\ Buffer\ Pool\ Size} = FEC\ Pool\ Size_{Channel\ 0} + \cdots + FEC\ Pool\ Size_{Channel\ last}$$

Receiver Output Data Control

Figure 3-2 illustrates the data rate pull out control.

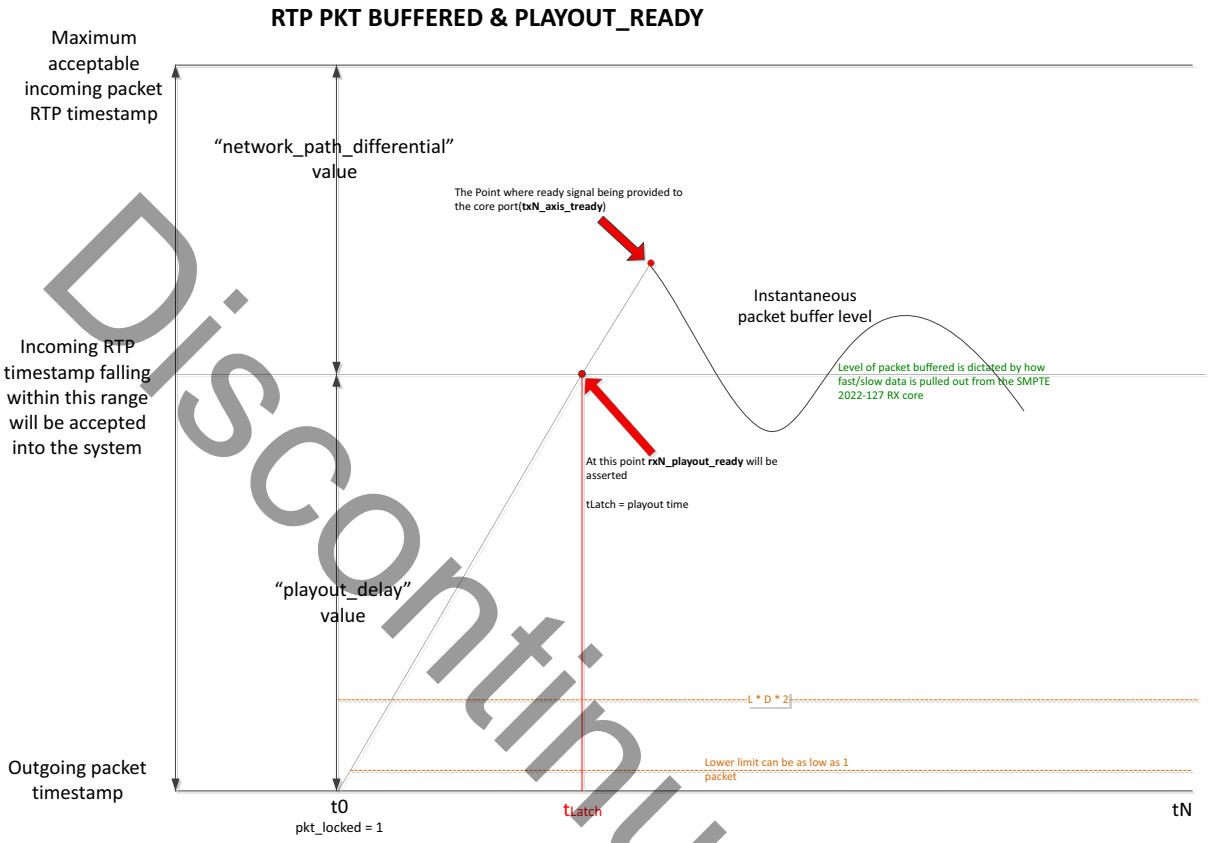

Figure 3-2: **Data Rate Pull out Control**

Data pull out rate from the SMPTE 2022-12 RX core is done by controlling AXIS tready signal into the RX core [txN_axis_tready]. This is done by monitoring two RX core status ports:

- rxN_playout_ready
- rxN_pkt_buffered

The txN_axis_tready is advised to be asserted as soon as the rxN_playout_ready goes high. This is also the instance when the rxN_pri_pkt_buffered equals the **playout_time** .

The ready txN_axis_tready should be asserted based on two criteria:

- rxN_pkt_buffered > $FEC_D * FEC_L * 2$
- rxN_pkt_buffered < value in **media_pkt_buf_size** register

AXI Memory Map Bandwidth Requirements

The memory bandwidth is calculated based on maximum input of 1Gbps per link to the SMPTE 2022-1/2 RX including RTP and FEC packet regardless of Channel Number, TS Rate, TS Size and TS per IP.

The values in Table 3-3 are based on worst case per port scenario.

Table 3-3: **Transmitter AXI-MM Port Bandwidth Consumption**

Port	Bandwidth (Gbps)
M0_AXIMM WR	1.95
M0_AXIMM RD	0.86
M1_AXIMM WR	0.20
M1_AXIMM RD	0.87

AXI Memory Map Latency

SMPTE 2022-1/2 Video over IP Receiver core has a 4-packet buffer at the input to hold incoming packets before they are written in the memory. In case of extreme long latency or insufficient bandwidth, the buffer overflows and may result in missing packets.

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 10]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 8]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 7]
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 5]

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 10] for detailed information. IP Integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values do change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl console.

Vivado Integrated Design Environment (IDE)

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

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

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

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

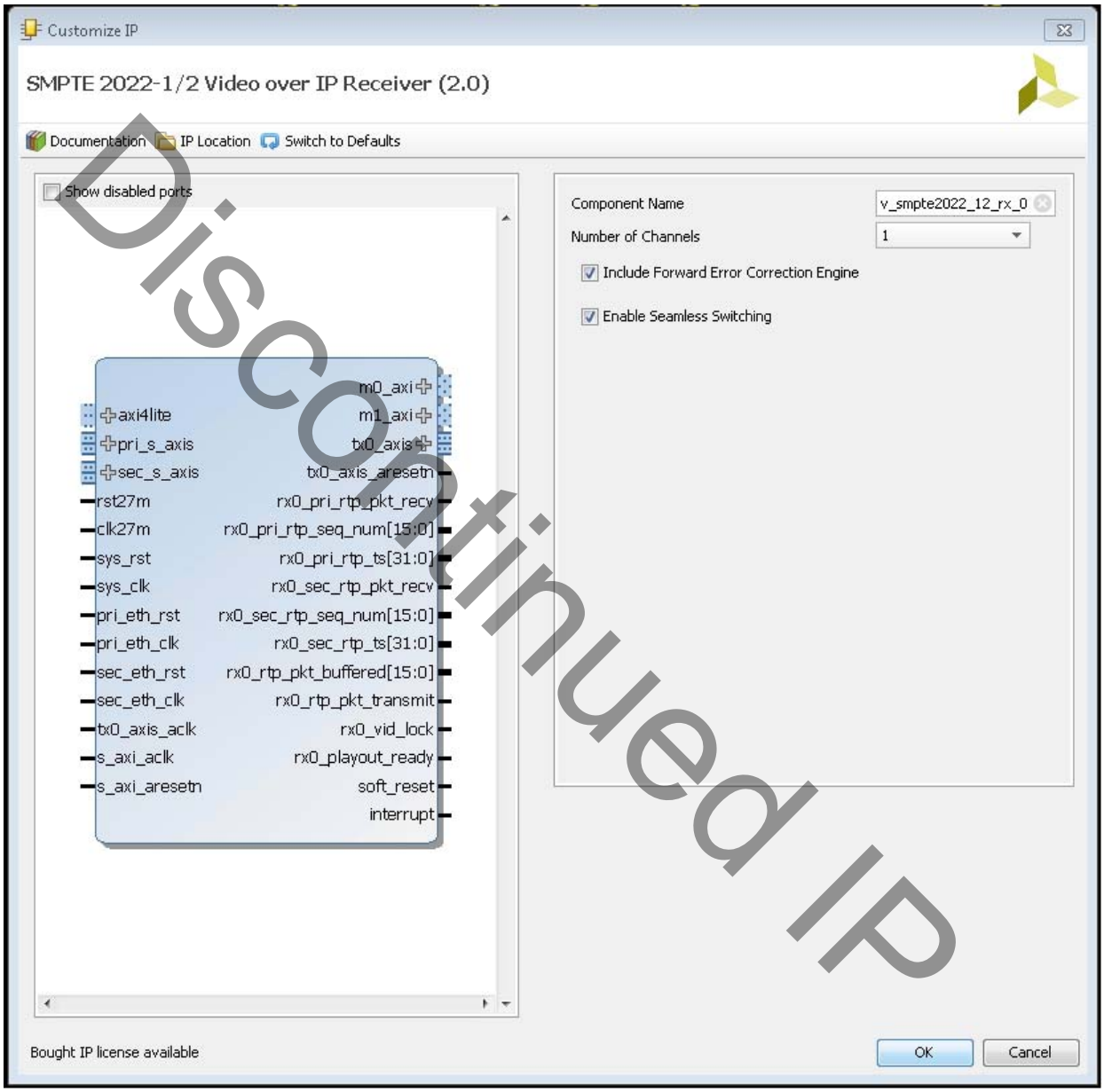

Figure 4-1: **SMPTE 2022-1/2 Video over IP Receiver Vivado Graphical User Interface**

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_smpte2022_12_rx_v1_0 cannot be used as a component name.
- **Number of Channels**: Specifies the number of channels.
- **Include Forward Error Correction Engine**: When checked, the core is generated with FEC.
- **Enable Seamless Switching**: When checked, the core is generated with Secondary AXIS Ethernet Link to support hitless operation.

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: **GUI Parameter to User Parameter Relationship**

GUI Parameter/Value[1]	User Parameter/Value[1]	Default Value
Number of Channels	C_CHANNELS	1
Include Forward Error Correction Engine	C_INCLUDE_FEC	1
Enable Seamless Switching	C_INCLUDE_HITLESS	1

1. Parameter values are listed in the table where the GUI parameter value differs from the user parameter value. Such values are shown in this table as indented below the associated parameter.

Output Generation

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

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 Placement

There is no specific clock placement requirement for this core.

Banking

There is no specific banking rule for this core.

I/O Standard and Placement

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

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 Placement

There is no specific clock placement requirement for this core.

Banking

There is no specific banking rule for this core.

I/O Standard and Placement

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

Simulation

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 5].

Synthesis and Implementation

For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 8].

Test Bench

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

As shown in Figure 5-1, the demonstration test bench is a simple Verilog module which configures and tests the SMPTE 2022-12 VoIP Receiver core. The test bench consists of several modules which generates the transport stream and IP packet and drives it to the core along with configuring the core and checking the data sanity of transport stream packet coming out of core. The transport stream driven to the core is of different size, length, rate and different matrix size across all the channels. The test bench is supplied as part of the Example Simulation output product group.

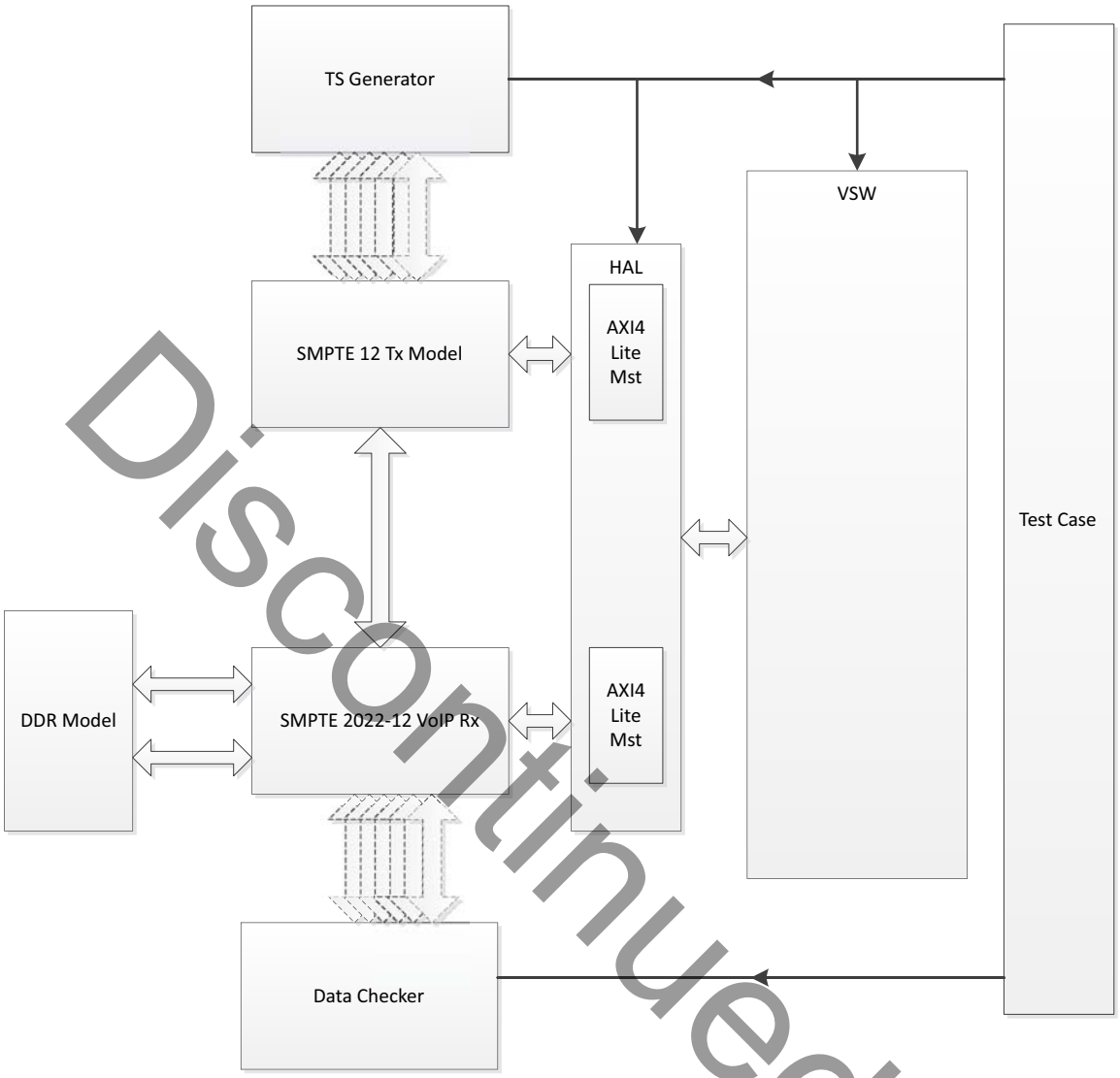

Figure 5-1: **SMPTE 2022-1/2 Video over IP Receiver Test Bench**

The main components of demonstration test bench are described below:

- **TS Generator**: This module generates Transport stream packets and drives it to SMPTE 12 Tx model across all the enabled channels.
- **SMPTE 12 Tx model**: This is SMPTE 2022-12 VoIP Transmitter model. This model receives TS packets converts it into IP packets and send it to SMPTE 2022-12 VoIP Receiver core.
- **HAL**: Hardware Access Layer is the register configuration layer. This layer has register read and write process.
- **VSW**: Virtual Software layer. 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 SMPTE 2022-1/2 Video over IP Receiver core has been validated using the Xilinx Kintex®-7 FPGA Broadcast Connectivity Kit. The core has also participated in various independent industry-led interoperability tests hosted by the Video Services Form (www.videoservicesforum.org).

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.

Migrating to the Vivado Design Suite

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

Upgrading in the Vivado Design Suite

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.

Parameter Changes

There are no parameter changes.

Port Changes

There are no port changes.

Other Changes

Added statistic and status registers.

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 may 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 SMPTE 2022-1/2 Video over IP Receiver, 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 SMPTE 2022-1/2 Video over IP Receiver. 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.

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 also 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

AR: 54532

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.

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

Debug Tools

There are many tools available to address SMPTE 2022-1/2 Video over IP Receiver 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 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).

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

IP Core Debug

Crucial Register Settings

1. Ensure bits [31:9] of network_path_differential (0x028) and bits [31:9] playout_delay (0x11C) registers are set accordingly.
2. Ensure media_pkt_buf_size (0x130) is large enough based on the computations described in Base Address and Play out Delay Register Setting in Chapter 3.
3. Ensure the fec_buf_base_addr (0x034) register and media_buf_base_addr (0x12C) register are set accordingly and not overlapping.
4. Ensure the match_sel (0x0B4) register is set properly to match the Ethernet Header (SSRC, IP Source, IP Destination, UDP Source, UDP Destination and VLAN TAG ID) which is set in the core and indicates which channel it belongs.

Debug Operation

1. If pri_recv_pkt_cnt (0x03C) register and sec_recv_pkt_cnt (0x040) register is incrementing, it indicates the core is receiving packets.
2. If pri_discard_pkt_cnt (0x04C) register and sec_discard_pkt_cnt (0x050) register is incrementing, it indicates the incoming packets is dropped due it does not match with the settings of match_sel (0x0B4) register.
3. Ensure the bit [0] of ts_status (0x120) register (packet lock) is high which indicates the packet lock to a right RTP packet based on the TS Size and TS per IP.
4. If link_valid_media_pkt_cnt (0x0C0) register and chan_valid_media_pkt_cnt (0x134) register is incrementing, it indicate valid packet is coming into the core.
5. If oor_pkts_cnt (0x160) register is incrementing, it indicates dropped packet due the incoming packet timestamp falls outside the window as set by bits [31:9] of network_path_differential (0x028) and bits [31:9] of playout_delay (0x11C) registers.

6. Ensure that the `curr_pkts_buffered` (0x140) is not greater than the `media_pkt_buf_size` (0x130) register.

Additional Resources 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. *Vivado AXI Reference Guide* (UG1037)
2. *SMPTE 2022-1/2 CBR MPEG2 Over IP with Forward Error Correction* (XAPP1194)
3. *LogiCORE IP AXI Interconnect Product Guide* (PG059)
4. *LogiCORE Tri-Mode Ethernet MAC Product Guide* (PG051)
5. *Vivado® Design Suite User Guide - Logic Simulation* (UG900)
6. *Vivado Design Suite User Guide - Implementation* (UG904)
7. *Vivado Design Suite User Guide: Getting Started* (UG910)
8. *Vivado Design Suite User Guide: Designing with IP* (UG896)
9. *ISE to Vivado Design Suite Migration Guide* (UG911)
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
10/05/2016	2.0	Updated Latency section. Updated Xilinx automotive applications disclaimer.
06/08/2016	2.0	Updated Ethernet Packet Received Interface and AXI4-Lite Register Map tables.
11/18/2015	2.0	Added UltraScale+ support.
04/01/2015	2.0	Added UltraScale™ support. Added core block diagram. Added AXI_MM port table. Updated AXI4-Lite registers. Added Test Bench chapter.
10/01/2014	2.0	Updated AXI4-Lite Register map and descriptions. Updated equations for Memory Requirements.
10/02/2013	1.0	Initial Xilinx release.

Please Read: Important Legal Notices