

S/PDIF v2.0

LogiCORE IP Product Guide

Vivado Design Suite

PG045 November 18, 2015

Table of Contents

IP Facts

Chapter 1: Overview

Functional Description.....	5
Feature Summary.....	10
Applications.....	10
Unsupported Features.....	10
Licensing and Ordering Information.....	11

Chapter 2: Product Specification

Standards.....	12
Performance.....	12
Resource Utilization.....	13
Port Descriptions.....	15
Register Space.....	18

Chapter 3: Designing with the Core

Clocking.....	25
Resets.....	26

Chapter 4: Design Flow Steps

Customizing and Generating the Core.....	27
Constraining the Core.....	30
Simulation.....	31
Synthesis and Implementation.....	31

Chapter 5: Example Design

Chapter 6: Test Bench

Test Bench Functionality.....	34
S/PDIF Core in RX Mode.....	34
S/PDIF Core in TX Mode.....	35

Appendix A: Migrating and Upgrading

Migrating to the Vivado Design Suite	36
Upgrading in the Vivado Design Suite	36

Appendix B: Debugging

Finding Help on Xilinx.com	37
Debug Tools	39
Hardware Debug	39
Interface Debug	40

Appendix C: Additional Resources and Legal Notices

Xilinx Resources	42
References	42
Revision History	43
Please Read: Important Legal Notices	44

Introduction

The Sony/Philips Digital Interconnect Format (S/PDIF) core is a digital audio interface controller that implements the International Electronic Commission (IEC) 60958-3 interface for transmitting and receiving audio data.

This includes standard bus interfaces to the AMBA® AXI4-Lite and AXI4-Stream interfaces [Ref 1], allowing for integration to the IP core with a master system for further processing of audio data. Data collected by the LogiCORE™ IP S/PDIF core is stored in the core internal FIFO, allowing the system to process a relatively slow audio stream.

Features

- Configurable as an S/PDIF audio data transmitter or an S/PDIF audio data receiver
- Configurable FIFO buffer stores the audio sample data

S/PDIF Interface

- IEC 60958-3 standard S/PDIF digital audio bus interface
- Two audio channels
- Audio sample lengths of 16/20/24 bits
- Data recovery from the bi-phase mark encoded S/PDIF data when the IP core is in receive mode
- Variable sampling rates (32/44.1/48/88.2/96/176.4/192 kHz)
- S/PDIF transmitter sends the invalid null audio frames over the S/PDIF line in case of a FIFO under-run condition

AXI4-Stream Interface

- Based on AXI4-Stream specification
- Master/slave on AXI4 streaming interface
- 32-bit data width support
- Continuous aligned streams only (no null or positional bytes transmission support)

AXI4-Lite Interface

- Register access support through the AXI4-Lite interface
- 32-bit data width support

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family ⁽¹⁾	UltraScale+™ Families, UltraScale™ Architecture, Zynq®-7000 All Programmable SoC, 7 Series
Supported User Interfaces	S/PDIF, AXI4-Stream, AXI4-Lite
Resources	See Table 2-2 to Table 2-4 .
Provided with Core	
Design Files	Encrypted HDL
Example Design	Verilog, VHDL
Test Bench	Verilog
Constraints File	XDC
Simulation Model	Verilog and VHDL Structural Models
Supported S/W Driver	Not Provided
Tested Design Flows ⁽²⁾	
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 listing of supported devices, see the Vivado IP catalog.
2. For the supported versions of the tools, see the [Xilinx Design Tools: Release Notes Guide](#).

Overview

Functional Description

The LogiCORE™ IP S/PDIF core is compatible with the S/PDIF protocol. It can be used in a receive or transmit mode and delivers or accepts audio data from an AXI4-Stream input. The S/PDIF core is designed for use in audio systems, and is used with the LogiCORE IP DisplayPort core for audio data transfers. [Figure 1-1](#) shows the S/PDIF block diagram.

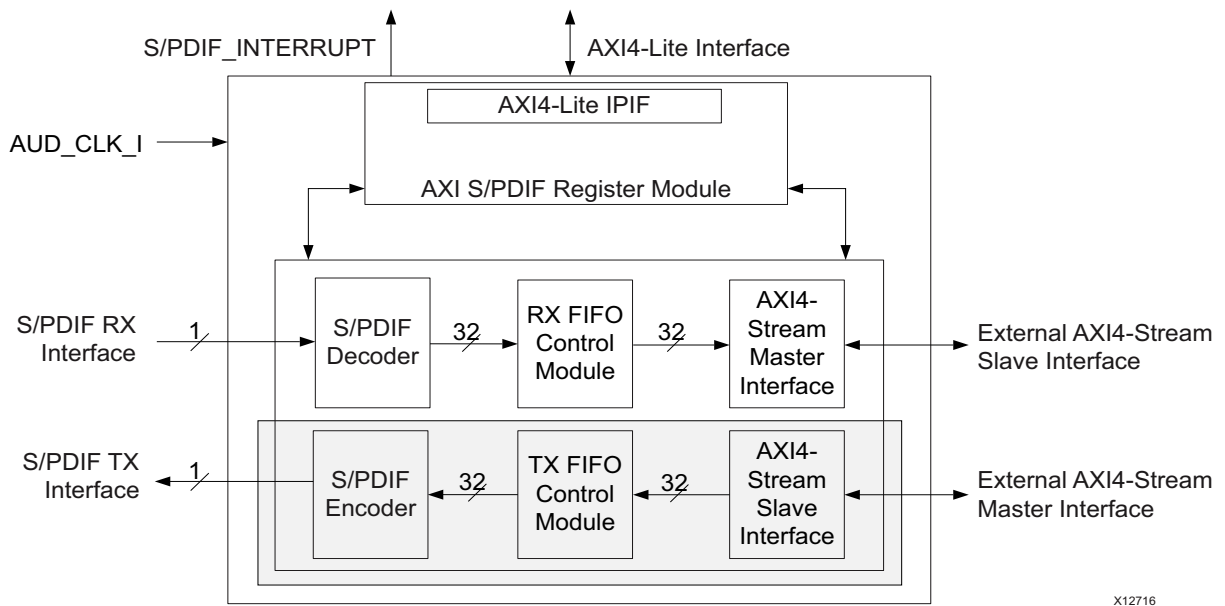


Figure 1-1: S/PDIF Block Diagram

The S/PDIF core can operate in two modes:

- **S/PDIF Receiver** – Receives S/PDIF audio data and sends it through the AXI4-Stream interface
- **S/PDIF Transmitter** – Receives audio data through the AXI4-Stream interface and transmits it through the S/PDIF interface

The core contains the following blocks in S/PDIF receive mode:

- AXI4-Lite IPIF
- AXI S/PDIF Register Module
- S/PDIF Decoder
- RX FIFO Control Module
- AXI4-Stream Master Interface

The core contains the following blocks in S/PDIF transmit mode:

- AXI4-Lite IPIF
- AXI S/PDIF Register Module
- AXI4-Stream Slave Interface
- TX FIFO Control Module
- S/PDIF Encoder

S/PDIF Register Module

This section describes the blocks within the S/PDIF register module.

AXI4-Lite IPIF

The S/PDIF register module provides the read/write control logic for the S/PDIF core register set. The registers are accessible by the AXI4-Lite master interface through the AXI4-Lite IPIF module, which is part of the S/PDIF register module.

The data width of the AXI4-Lite interface is fixed at 32 bits. The registers are defined in [Table 2-6, page 18](#).

The interrupt control and soft reset functionality are also implemented as part of the S/PDIF register module. The S/PDIF core can be reset by writing `0xA` to the soft reset register. For an S/PDIF transmitter, an S/PDIF interrupt can be generated based on the FIFO Full/FIFO empty conditions. For an S/PDIF receiver, in addition to the FIFO Full/FIFO Empty conditions, the interrupt is triggered if any preamble error/bi-phase mark code (BMC) error is detected over the S/PDIF line or if the S/PDIF core receives the start of block over the S/PDIF line.

S/PDIF Decoder

The enable bit in the Control register of the S/PDIF register module has to be set to enable the S/PDIF decoder module. The S/PDIF decoder recovers data from the bi-phase mark coded S/PDIF data stream (see [Bi-Phase Mark Code](#)). The audio clock frequency should be at least eight times the bit rate $64 \times FS$. FS is the sampling frequency where each sample has 64 bits. In bi-phase mark code, each bit changes twice in a bit period.

For example, to recover data from a 192 kHz sampling rate, the minimum audio clock frequency should be $8 \times 64 \times 192 \text{ kHz} = 98.304 \text{ MHz}$ to recover the data samples. As per the S/PDIF Protocol Preamble violates the bi-phase mark code format, S/PDIF decoder module identifies the channel number and the start of the audio block from the Preamble pattern.

The serial-to-parallel data conversion also takes place in the S/PDIF Decoder module and then generates the FIFO write enables with the 32-bit FIFO input data. The sampling frequency information (that is, the count of audio clocks during the bit period) is updated in the Status register of the S/PDIF register module.

This module detects the BMC/Preamble errors over the S/PDIF line and reports to the S/PDIF register module.

RX FIFO Control Module

The Asynchronous RX FIFO is used to store the 32-bit audio data received from the S/PDIF decoder. The FIFO size is configurable and based on the C_AXIS_BUFFER_SIZE parameter generated. The data width of the FIFO is fixed to 32 bits. This module receives the FIFO write input control and write data from the S/PDIF decoder. When the FIFO reaches the full condition, an interrupt is generated and the RX FIFO Full status is updated through the Status register of the S/PDIF register module. Similarly, the RX FIFO empty interrupt and corresponding status are updated through the S/PDIF Interrupt Status register.

AXI4-Stream Interface

The data width over the AXI4-Stream interface is fixed at 32 bits. [Table 1-1](#) shows the 32-bit data format over the AXI4-Stream interface during audio data transmission and reception. All bit positions are as per the IEC60958-3 standard except for the preamble bit format.

Table 1-1: AXI4-Stream Audio Data Format

Bit[31]	Bit[30]	Bit[29]	Bit[28]	Bits[27:4]	Bits[3:0]
Parity	Channel Status Bit	User Data Bit	Validity Bit	Audio Sample Data	Preamble

The preamble provides the start of the audio block and audio channel information. The preamble patterns for the start of block, channel1 audio data, and channel2 audio data are listed in [Table 1-2](#).

Table 1-2: Preamble

Bits[3:0]	Description
0001	Start of audio block
0010	Channel1 audio data
0011	Channel2 audio data

Bits[27:4] carry the audio data MSB bit at the 27th position and the LSB position is based on the audio sample length. Bit[28] provides the audio validity information. Bit[29] carries the user data information, and Bit[30] carries the channel status bit. Bit[31] is the even parity over 32 bits except for the preamble bits.

AXI4-Stream Master Interface

The AXI4-Stream Master interface transfers the 32-bit parallel data read from the non-empty FIFO to the AXI4-Stream interface. The corresponding data valid signal (`m_axis_tvalid`) is set and the channel identifier signal (`m_axis_tid`) is driven with the corresponding channel number. The channel number information is available to the AXI4-Stream Master interface through the S/PDIF decoder. This module depends on the handshaking signal `m_axis_tready` issued from the AXI4-Stream interface target slave for completion of the transfer.

AXI4-Stream Slave Interface

The AXI4-Stream Slave interface receives the 32-bit streaming data from the target connected to the AXI4-Stream interface. This module generates the handshaking signal `s_axis_tready` after receiving the streaming data (`s_axis_tdata`), data valid signal (`s_axis_tvalid`), and channel number identification (`s_axis_tid`). This also generates the TX FIFO write control signals and transfers the data received from the AXI4-Stream interface to the TX FIFO Control Module. If the S/PDIF TX FIFO is full, this module stops receiving the audio samples by driving the handshake signal `s_axis_tready` Low. This avoids the FIFO overrun condition in the S/PDIF transmitter.

TX FIFO Control Module

The Asynchronous TX FIFO is used to store the 32-bit streaming data received from the AXI4-Stream slave interface. The FIFO size is user configurable. The data width of the FIFO is fixed at 32 bits.

This module receives the FIFO write input control and write data from the AXI4-Stream slave interface. When the FIFO reaches the full condition, an interrupt is generated and the TX FIFO Full status is updated through the Interrupt Status register of the S/PDIF register module. This condition can occur when the S/PDIF enable bit is not set in the Control register of the S/PDIF register module, and the target is sending continuous streaming data or when the FIFO Size is not sufficient.

The TX FIFO Control Module generates a TX_FIFO_EMPTY interrupt when the TX FIFO becomes empty. This condition can occur if enough samples are not received by the S/PDIF transmitter to send over the S/PDIF line.

S/PDIF Encoder

This module has to get the clock configuration bits through the Control register to know the bit rate before S/PDIF data transmission starts. The enable bit in the Control register of the S/PDIF registers module must be set to enable the S/PDIF encoder module. The S/PDIF encoder converts the 32-bit parallel data received from the TX FIFO to serial data. The serial data is transferred over the S/PDIF interface in bi-phase mark code (BMC) format with respect to the received bit rate information. The audio clock input must be the harmonic of the sampling rate and should be higher than the bit rate. For example, for a 192 kHz sampling rate, 49.152 MHz or 98.3 MHz must be provided as the core frequency (AUD_CLK_I) and the corresponding clock divisor information must be given through the clock configuration register bits).

In case of a TX FIFO under-run condition, this module sends the null audio frames over the S/PDIF line with the validity bit set. When the validity bit is set to 1, it means per the S/PDIF protocol that the audio sample is invalid and the codec has to ignore the sample.



IMPORTANT: *The audio clock generation and the setting of clock configuration bits in the Control register must be done with care to support these sampling rates: 32 kHz, 44.1 kHz, 48 kHz, 88.2 kHz, 96 kHz, 176.4 kHz, and 192 kHz. The minimum audio clock frequency of 49.152 MHz or harmonic frequency of the same which is higher than this frequency supports the sampling rates of 32 kHz, 48 kHz, 96 kHz, and 192 kHz. The minimum audio clock frequency of 45.1584 MHz or harmonic frequency of the same which is higher than this frequency supports the sampling rates of 44.1 kHz, 88.2 kHz, and 176.4 kHz.*

Bi-Phase Mark Code

The S/PDIF interface (IEC-60958) is a consumer version of the AES/EBU-interface. The S/PDIF digital signal is coded using the bi-phase mark code, which is a type of phase modulation. The bit clock, data bits, and BMC signals are shown in Figure 1-2.

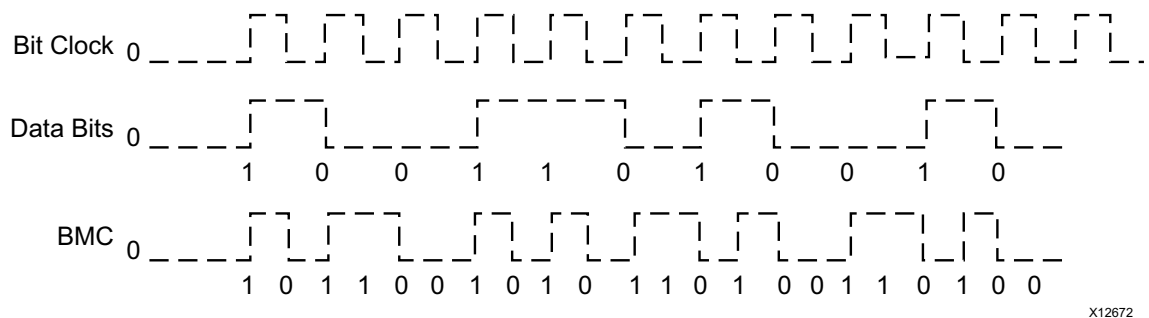


Figure 1-2: Bi-Phase Mark Code

The frequency of the clock is twice the bit rate. Every bit of the original data is represented as two logical states, which, together, form a cell. The length of a cell (time slot) is equal to the length of a data bit. The logical level at the start of a bit is always inverted to the level at the end of the previous bit. The level at the end of a bit is equal (a 0 is transmitted) or inverted (a 1 is transmitted) to the start of that bit. The BMC has either one or two transitions for every bit. Data bit 1 has two transitions during a bit period, and Data bit 0 has one transition during a bit period. As per protocol, except for the preambles, S/PDIF audio data is transferred in the BMC format. Preambles violate the BMC to identify the channel information by the S/PDIF receivers.

Feature Summary

The LogiCORE IP S/PDIF core is a digital audio interface controller that implements the PCM IEC 60958-3 interface features for transmitting and receiving audio data. The core can be configured as an S/PDIF audio data transmitter or an S/PDIF audio data receiver. The IEC 60958-3 standard S/PDIF digital audio bus interface has two audio channels and audio sample lengths of 16, 20, and 24 bits. Sample rates range from 32 kHz to 192 kHz.

The core includes an AMBA[®] AXI4-Lite interface for register access and an AXI4-Stream interface for audio data transfers. The AXI4-Stream interface allows integration between the IP core and an AXI system for further processing of audio data. Data collected by the S/PDIF core is stored in the core internal FIFO, allowing the system to process a relatively slow audio stream.

Applications

The S/PDIF core can be used as a short distance audio interconnect. It can be used to carry digital audio as defined by the standard IEC60958-3.

Unsupported Features

The S/PDIF core does not support non-linear PCM encoded audio data streams.

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 DisplayPort [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 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

Standards

The LogiCORE™ IP S/PDIF core implements IEC 60958-3 interface features for transmitting and receiving audio data.

Performance

Performance characterization of this core has been done using margin system methodology. The details of the margin system characterization methodology is described in the *Vivado Design Suite User Guide, Designing With IP* (UG896) [Ref 3].

Note: Maximum frequency numbers for UltraScale™ architecture and Zynq®-7000 devices are expected to be similar to 7 series device numbers.

Table 2-1: Maximum Frequencies

Family	Speed Grade	F _{Max} (MHz)	
		AXI4-Lite	AXI4-Stream
Virtex-7	-1	180	200
Kintex-7		180	200
Artix-7		120	150
Virtex-7	-2	200	240
Kintex-7		200	240
Artix-7		140	180
Virtex-7	-3	220	280
Kintex-7		220	280
Artix-7		160	200

Resource Utilization

Note: Resource numbers for UltraScale™ architecture and Zynq®-7000 devices are expected to be similar to 7 series device numbers.

Virtex-7 FPGAs

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

Table 2-2: Device Utilization – Virtex-7 FPGAs

Parameter Values				Device Resources			
S/PDIF Mode	FIFO Depth	Enable Channel Status Registers	Enable User Data Registers	Slices	Slice Flip-Flops	LUTs	Block RAM
Receiver	16	0	0	96	221	192	1
Receiver	512	1	1	105	296	228	1
Transmitter	16	–	–	159	359	339	1
Transmitter	512	–	–	181	434	374	1

Kintex-7 FPGAs

Table 2-3 provides approximate resource counts for the various core options using Kintex®-7 FPGAs.

Table 2-3: Device Utilization – Kintex-7 FPGAs

Parameter Values				Device Resources			
S/PDIF Mode	FIFO Depth	Enable Channel Status Registers	Enable User Data Registers	Slices	Slice Flip-Flops	LUTs	Block RAM
Receiver	16	0	0	101	221	192	1
Receiver	512	1	1	115	296	229	1
Transmitter	16	–	–	155	359	339	1
Transmitter	512	–	–	169	434	374	1

Artix-7 FPGAs

Table 2-4 provides approximate resource counts for the various core options using Artix®-7 FPGAs.

Table 2-4: Device Utilization – Artix-7 FPGAs

Parameter Values				Device Resources			
S/PDIF Mode	FIFO Depth	Enable Channel Status Registers	Enable User Data Registers	Slices	Slice Flip-Flops	LUTs	Block RAM
Receiver	16	0	0	105	221	192	1
Receiver	512	1	1	105	296	229	1
Transmitter	16	–	–	160	359	339	1
Transmitter	512	–	–	164	434	374	1

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.

Resources required for the S/PDIF core have been estimated in Table 2-2 to Table 2-4. These values were generated using the Vivado IP catalog. They are derived from post-synthesis reports, and might change during implementation.

Start by choosing the Family, Maximum Frame Size, and Minimum Block size of the core.

Port Descriptions

The S/PDIF Input/Output (I/O) signals are listed and described in [Table 2-5](#).

Table 2-5: I/O Signal Description

Signal Name	Interface	I/O	Initial State	Description
System Signals				
aud_clk_i	System	I	–	Audio clock input used at the S/PDIF interface
spdif_interrupt	System	O	0	S/PDIF core interrupt output. When the interrupt occurs, this signal is continuously 1 until cleared/disabled.
AXI4-Lite Interface System Signals				
s_axi_aclk	System	I	–	AXI4-Lite clock
s_axi_aresetn	System	I	–	AXI4-Lite reset, active-Low
AXI4-Lite Write Address Channel Signals				
s_axi_awaddr[c_s_axi_addr_width – 1:0]	AXI4-Lite	I	–	AXI4-Lite Write address. The write address bus gives the address of the first transfer in a write burst transaction.
s_axi_awvalid	AXI4-Lite	I	–	Write address valid. This signal indicates that valid write address and control information are available.
s_axi_awready	AXI4-Lite	O	0	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
AXI4-Lite Write Data Channel Signals				
s_axi_wdata[c_s_axi_data_width – 1:0]	AXI4-Lite	I	–	Write data bus
s_axi_wstrb[c_s_axi_data_width/8 – 1:0]	AXI4-Lite	I	–	Write strobes. Each signal indicates which byte lanes to update in memory. These are unused in the S/PDIF core.
s_axi_wvalid	AXI4-Lite	I	–	Write valid. This signal indicates that valid write data and strobes are available.
s_axi_wready	AXI4-Lite	O	0	Write ready. This signal indicates that the slave can accept the write data.
AXI4-Lite Write Response Channel Signals				
s_axi_bresp[1:0]	AXI4-Lite	O	0	Write response. This signal indicates the status of the write transaction.
s_axi_bvalid	AXI4-Lite	O	0	Write response valid. This signal indicates that a valid write response is available.
s_axi_bready	AXI4-Lite	I	–	Response ready. This signal indicates that the master can accept the response information.

Table 2-5: I/O Signal Description (Cont'd)

Signal Name	Interface	I/O	Initial State	Description
AXI4-Lite Read Address Channel Signals				
s_axi_araddr[c_s_axi_addr_width – 1:0]	AXI4-Lite	I	–	Read address. The read address bus gives the initial address of a read burst transaction.
s_axi_arvalid	AXI4-Lite	I	–	Read address valid. When High, this signal indicates that the read address and control information are valid and remain stable until the address acknowledgment signal s_axi_arready is High.
s_axi_arready	AXI4-Lite	O	0	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals.
AXI4-Lite Read Data Channel Signals				
s_axi_rdata[c_s_axi_data_width – 1:0]	AXI4-Lite	O	0	Read data bus
s_axi_rresp[1:0]	AXI4-Lite	O	0	Read response. This signal indicates the status of the read transfer.
s_axi_rvalid	AXI4-Lite	O	0	Read valid. This signal indicates that the required read data is available and the read transfer can complete.
s_axi_rready	AXI4-Lite	I	–	Read ready. This signal indicates that the master can accept the read data and response information.
S/PDIF RX Interface Signals				
spdif_i	S/PDIF	I	–	Audio input from the S/PDIF interface
S/PDIF TX Interface Signals				
spdif_o	S/PDIF	O	0	Audio output to the S/PDIF interface
AXI4-Stream Master Interface Signals				
m_axis_aclk	M_AXISStream	I	–	The AXI4-Stream global clock signal in receive mode. All streaming signals are sampled on the rising edge of m_axis_aclk.
m_axis_aresetn	M_AXISStream	I	–	The AXI4-Stream global reset signal in receive mode. m_axis_aresetn is active-Low.
m_axis_tvalid	M_AXISStream	O	0	AXI4-Stream Valid Out. Indicates stream data bus, m_axis_tdata, is valid. 0 = Write data is not valid 1 = Write data is valid
m_axis_tready	M_AXISStream	I	–	AXI4-Stream Ready. Indicates to the AXI4-Stream Master interface that the target is ready to receive stream data. 0 = Not ready to receive data 1 = Ready to receive data
m_axis_tdata[c_axis_tdata_width – 1:0]	M_AXISStream	O	0	AXI4-Stream Data Out

Table 2-5: I/O Signal Description (Cont'd)

Signal Name	Interface	I/O	Initial State	Description
m_axis_tid[c_axis_tid_width – 1:0]	M_AXISStream	O	0	m_axis_tid is the data stream identifier that indicates channel number of audio data.
AXI4-Stream Slave Interface Signals				
s_axis_aclk	S_AXISStream	I	–	The AXI4-Stream global clock signal in transmit mode. All signals are sampled on the rising edge of s_axis_aclk.
s_axis_aresetn	S_AXISStream	I	–	The AXI4-Stream global reset signal in transmit mode. s_axis_aresetn is active-Low.
s_axis_tvalid	S_AXISStream	I	–	AXI4-Stream Valid In. Indicates the stream data bus, s_axis_tdata, is valid. 0 = Write data is not valid 1 = Write data is valid
s_axis_tready	S_AXISStream	O	0	AXI4-Stream Ready. Indicates the AXI4-Stream Slave interface is ready to receive stream data. 0 = Not ready to receive data 1 = Ready to receive data
s_axis_tdata[c_axis_tdata_width – 1:0]	S_AXISStream	I	–	AXI4-Stream Data In
s_axis_tid[c_axis_tid_width – 1:0]	S_AXISStream	I	–	s_axis_tid is the data stream identifier that indicates channel number of audio data.

Register Space

Table 2-6 specifies the offset address, register name, and accessibility of each firmware addressable register from the three classes of registers within the S/PDIF core. User access to each register is from an offset to the base address.

For Channel Status registers, capturing channel status bits into channel status registers is configurable. When the Channel Status registers are enabled, only these registers are part of S/PDIF receiver logic. Channel Status registers hold the 192-bit channel status information received over the S/PDIF input when the S/PDIF core is in receive mode. The channel status is assumed to be common for both channel a and channel b. Thus the channels status bits are captured from one of the channels. These registers are updated after one complete audio frame is received. Usually, the channel status register data does not change frame to frame. For more information on these bits including their descriptions, see the IEC-60958-3 specification.

For the Channel a/b User Data registers, capturing S/PDIF user data bits into user data registers is configurable. When the User Enabled registers are enabled, only these registers are part of the S/PDIF receiver logic. User Data registers hold the 192-bit user data received over the S/PDIF input when the S/PDIF core is in receive mode. The user data is captured for both channel a and channel b in the corresponding registers. These registers are updated after one complete audio frame is received.

Table 2-6: Register Map

Offset Address (Hex)	Register Name	Description
Interrupt Registers		
0x1C	Global Interrupt Enable (GIE) ⁽¹⁾	Device Global interrupt enable register
0x20	Interrupt Status Register (ISR) ⁽¹⁾	IP interrupt status register
0x28	Interrupt Enable Register (IER) ⁽¹⁾	IP interrupt enable register
Soft Reset Register		
0x40	Soft Reset Register ⁽²⁾	Soft Reset Register
S/PDIF Configuration, Control, and Data Registers		
0x44	S/PDIF Control Register	Control register
0x48	S/PDIF Status Register	Status register
0x4C	Channel Status Register0	Audio Channel status bits 0 to 31
0x50	Channel Status Register1	Audio Channel status bits 32 to 63
0x54	Channel Status Register2	Audio Channel status bits 64 to 95
0x58	Channel Status Register3	Audio Channel status bits 96 to 127
0x5C	Channel Status Register4	Audio Channel status bits 128 to 159

Table 2-6: Register Map (Cont'd)

Offset Address (Hex)	Register Name	Description
0x60	Channel Status Register5	Audio Channel status bits 160 to 191
0x64	Channel A User Data Register0	Channel A User Data bits 0 to 31
0x68	Channel A User Data Register1	Channel A User Data bits 32 to 63
0x6C	Channel A User Data Register2	Channel A User Data bits 64 to 95
0x70	Channel A User Data Register3	Channel A User Data bits 96 to 127
0x74	Channel A User Data Register4	Channel A User Data bits 128 to 159
0x78	Channel A User Data Register5	Channel A User Data bits 160 to 191
0x7C	Channel B User Data Register0	Channel B User Data bits 0 to 31
0x80	Channel B User Data Register1	Channel B User Data bits 32 to 63
0x84	Channel B User Data Register2	Channel B User Data bits 64 to 95
0x88	Channel B User Data Register3	Channel B User Data bits 96 to 127
0x8C	Channel B User Data Register4	Channel B User Data bits 128 to 159
0x90	Channel B User Data Register5	Channel B User Data bits 160 to 191

Notes:

1. See the *Xilinx Interrupt Control Data Sheet* [Ref 4].
2. The soft reset functionality is implemented by the soft_reset module.

Global Interrupt Enable (GIE)

The Global Interrupt Enable register, described in [Table 2-7](#), has a single defined bit that globally enables the final interrupt out to the system.

Table 2-7: Global Interrupt Enable Register (Offset 0x1C)

Bits	Name	Reset Value	Access Type	Description
31	GIE	0	R/W	Global Interrupt Enable 0 = All interrupts disabled. No interrupts from S/PDIF 1 = Unmasked S/PDIF interrupts are passed to the processor
30:0	Unused	N/A	N/A	Reserved

Interrupt Status Register (ISR)

Firmware uses the ISR to determine which interrupt events from the S/PDIF core need servicing. Writing a 1 to a bit position within the register causes the corresponding bit to toggle. All register bits are cleared upon reset. The register uses a toggle on write method to allow the firmware to clear selected interrupts by writing a 1 to the desired interrupt bit field position.

This mechanism avoids the requirement on the User Interrupt Service routine to perform a Read/Modify/Write operation to clear a single bit within the register. An interrupt value of 1 indicates the interrupt has occurred. A value of 0 indicates that no interrupt occurred or it was cleared. The Interrupt Status register bit fields are described in [Table 2-8](#).

Table 2-8: Interrupt Status Register (Offset 0x20)

Bits	Name	Reset Value	Access Type	Description
31:5	Unused	0	N/A	Reserved
4	Preamble Error	0	Read/Toggle on writing 1	This bit is set when the incorrect preamble format is received over the S/PDIF core in receive mode, for example, if the channel a preamble is received after the start of block.
3	BMC Error	0	Read/Toggle on writing 1	This bit is set when there is a bi-phase mark code (BMC) violation over the S/PDIF audio data bits in receive mode (except for the preamble).
2	Start Of Block	0	Read/Toggle on writing 1	This bit is set when the S/PDIF core is in receive mode and when it detects the start of block preamble over the S/PDIF_I input.
1	TX/RX FIFO Empty	0	Read/Toggle on writing 1	This bit is set when the TX FIFO changes from non-empty to empty in Transmit mode and when the RX FIFO changes from non-empty to empty.
0	TX/RX FIFO Full	0	Read/Toggle on writing 1	This bit is set when the TX FIFO becomes full in transmit mode and when the RX FIFO becomes full in receive mode.

Interrupt Enable Register

The Interrupt Enable register is a read and write register that enables the S/PDIF interrupts. The Interrupt Enable register bit fields are described in [Table 2-9](#).

Table 2-9: Interrupt Enable Register (Offset 0x28)

Bits	Name	Reset Value	Access Type	Description
31:5	Unused	N/A	N/A	Reserved
4	Preamble Error	0	R/W	This bit must be set to generate the preamble error interrupt. In transmit mode, this bit is unused.

Table 2-9: Interrupt Enable Register (Offset 0x28) (Cont'd)

Bits	Name	Reset Value	Access Type	Description
3	BMC Error Interrupt Enable	0	R/W	This bit must be set to generate the BMC error interrupt. In transmit mode, this bit is unused.
2	Start Of Block Interrupt Enable	0	R/W	This bit must be set to generate the start of block interrupt in receive mode. In transmit mode, this bit is unused.
1	TX/RX FIFO Empty Interrupt Enable	0	R/W	This bit must be set to generate the TX FIFO empty interrupt when the S/PDIF core is in transmit mode and the same bit must be set when the S/PDIF core is in transmit mode to enable the RX FIFO empty interrupt.
0	TX/RX FIFO Full Interrupt Enable	0	R/W	This bit must be set to generate the TX FIFO full interrupt when the S/PDIF core is in transmit mode and the same bit has to be set when the S/PDIF core is in receive mode to enable the RX FIFO full interrupt.

Soft Reset Register

The firmware writes to the Soft Reset register to initialize all of the S/PDIF registers to their default states. To accomplish this, the firmware must write the value of 0xA to the least-significant nibble of the 32-bit word. After recognizing a write of 0xA, the soft_reset module issues a pulse four clocks long to reset the S/PDIF core. At the end of the pulse, the Soft Reset register acknowledges the AXI4 transaction, which prevents anything further from happening while the reset occurs. Writing any value to Bits[3:0] other than 0xA results in an AXI4 transaction acknowledge with an error status. This register is not readable. The Soft Reset register bit fields are described in [Table 2-10](#).

Table 2-10: Soft Reset Register (Offset 0x40)

Bits	Name	Reset Value	Access Type	Description
31:4	Unused	N/A	N/A	Reserved
3:0	Reset Key	0	W	The firmware must write a value of 0xA to this field to cause a soft reset of the interrupt registers of the S/PDIF controller. Writing any other value results in an AXI4 transaction acknowledgment with SLVERR and no reset occurs.

S/PDIF Control Register

The S/PDIF Control register is a read and write register that configures the S/PDIF core. This register has an S/PDIF enable bit, a TX/RX FIFO flush bit, and clock configuration bits. The S/PDIF Control register bit fields are described in [Table 2-11](#).

Table 2-11: S/PDIF Control Register (Offset 0x44)

Bits	Name	Reset Value	Access Type	Description
31:6	Unused	N/A	N/A	Reserved
5:2	TX clock Configuration Bits	0	R/W	These bits provide the audio clock division number to transmit the S/PDIF bits. The bit frequency is generated based on these bits. For example, to generate a 32 kHz audio sampling frequency, the bit rate is 2.048 MHz (that is, 32 kHz × 64 because the audio sample width is of 64 bits, 32 bits for channel a and 32 bits for channel b). If the supplied AUD_CLK_I is 16.384, the Bits Division Number has to be 0001. Bits Division Number: 0000 = 4 0001 = 8 0010 = 16 0011 = 24 0100 = 32 0101 = 48 0110 = 64 Others = Reserved
1	S/PDIF TX FIFO/RX FIFO Flush	0	R/W	This bit must be set to 1 to reset the TX FIFO in transmit mode and to reset the RX FIFO in receive mode.
0	S/PDIF TX/RX Enable	0	R/W	This bit must be set to 1 to enable the S/PDIF core.

S/PDIF Status Register

The S/PDIF Status register is a read-only register that contains the status of the S/PDIF core. The S/PDIF Status register bit fields are described in [Table 2-12](#).

Table 2-12: S/PDIF Status Register (Offset 0x48)

Bits	Name	Reset Value	Access Type	Description
31:10	Unused	N/A	N/A	Reserved
9:0	Sample clock count	0	R	These bits are updated with the number of audio clocks for the S/PDIF data bit period. This audio clock count is recovered by the S/PDIF decoder module. This count gives the approximate count for the S/PDIF bit period when the audio clock is not the harmonic of core frequency. These bits are used in receive mode only. In transmit mode, these bits are unused.

Channel Status Registers

A set of six configurable registers store the 192-bit S/PDIF Audio Channel Status information. These registers are active when the S/PDIF core is in receive mode and when the Enable Channel Status registers are 1. This channel status information is captured from one of the channels, assuming both channel a and channel b carry the same channel status information over S/PDIF. The Channel Status register bit fields are described in [Table 2-13](#). For complete descriptions of these bit fields, see the IEC-60958-3 specification.

Table 2-13: Channel Status Registers (Offsets 0x4C to 0x60)

Bits	Name	Reset Value	Access Type	Description
31:0	Channel Status Register0	0	R	This register holds bits 0 to 31 of the audio channel status information received over S/PDIF.
31:0	Channel Status Register1	0	R	This register holds bits 32 to 63 of the audio channel status information received over S/PDIF.
31:0	Channel Status Register2	0	R	This register holds bits 64 to 95 of the audio channel status information received over S/PDIF.
31:0	Channel Status Register3	0	R	This register holds bits 96 to 127 of the audio channel status information received over S/PDIF.
31:0	Channel Status Register4	0	R	This register holds bits 128 to 159 of the audio channel status information received over S/PDIF.
31:0	Channel Status Register5	0	R	This register holds bits 160 to 191 of the audio channel status information received over S/PDIF.

Channel A User Data Registers

A set of six configurable registers store the 192-bit S/PDIF Channela User data information. These registers are active when the S/PDIF core is in receive mode and when the Enable User Data registers are 1. This user data information is captured from channel a. The Channel A User Data registers bit fields are described in [Table 2-14](#).

Table 2-14: Channel A User Data Registers (Offsets 0x64-0x78)

Bits	Name	Reset Value	Access Type	Description
31:0	Channela User Data Register0	0	R	This register holds bits 0 to 31 of the user data information received over S/PDIF channel a.
31:0	Channela User Data Register1	0	R	This register holds bits 32 to 63 of the audio user data information received over S/PDIF channel a.
31:0	Channela User Data Register2	0	R	This register holds bits 64 to 95 of the user data information received over S/PDIF channel a.
31:0	Channela User Data Register3	0	R	This register holds bits 96 to 127 of the user data information received over S/PDIF channel a.

Table 2-14: Channel A User Data Registers (Offsets 0x64-0x78) (Cont'd)

Bits	Name	Reset Value	Access Type	Description
31:0	Channela User Data Register4	0	R	This register holds bits 128 to 159 of the user data information received over S/PDIF channel a.
31:0	Channela User Data Register5	0	R	This register holds bits 160 to 191 of the user data information received over S/PDIF channel a.

Channel B User Data Registers

A set of six configurable registers store the 192-bit S/PDIF Channelb User data information. These registers are active when the S/PDIF core is in receive mode and when the Enable User Data registers are 1. This user data information is captured from channel b. The Channel B User data registers bit fields are described in [Table 2-15](#).

Table 2-15: Channel B User Data Registers (Offsets 0x7C to 0x90)

Bits	Name	Reset Value	Access Type	Description
31:0	Channelb User Data Register0	0	R	This register holds bits 0 to 31 of the user data information received over S/PDIF channel b.
31:0	Channelb User Data Register1	0	R	This register holds bits 32 to 63 of the audio user data information received over S/PDIF channel b.
31:0	Channelb User Data Register2	0	R	This register holds bits 64 to 95 of the user data information received over S/PDIF channel b.
31:0	Channelb User Data Register3	0	R	This register holds bits 96 to 127 of the user data information received over S/PDIF channel b.
31:0	Channelb User Data Register4	0	R	This register holds bits 128 to 159 of the user data information received over S/PDIF channel b.
31:0	Channelb User Data Register5	0	R	This register holds bits 160 to 191 of the user data information received over S/PDIF channel b.

Designing with the Core

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

Clocking

The S/PDIF core interfaces with three different clocks which are described as follows. Asynchronous FIFOs are used for the audio data cross clocking paths.

- **aud_clk_i** – The S/PDIF audio interface works with this clock. The frequency of the clock should be ≥ 512 times the audio sampling rate for S/PDIF in receive mode.

For S/PDIF in transmit mode, the frequency of this clock should be a harmonic of the audio sampling rate and the corresponding divisor should be set in the S/PDIF control register. For example, a 32 kHz sampling rate, 16.384 MHz is the minimum **aud_clk_i** rate required.

- **s_axi_aclk** – This is the processor domain. The AXI4-Lite interface to the S/PDIF register access works with this clock.
- **s_axis_aclk/m_axis_aclk** – This is the audio streaming interface clock. This clock should be ≥ 512 times the sampling rate to match the audio rates.

Table 3-1: S/PDIF Clock Frequencies

Clock	Min	Max	Description
aud_clk_i	16 MHz	100 MHz	Audio clock input
s_axi_aclk	25 MHz	135 MHz	Host processor clock
s_axis_aclk/m_axis_aclk	16 MHz	100 MHz	Audio streamlining interface clock

Resets

The S/PDIF core uses the following resets.

- AXI4-Lite interface – `s_axi_aresetn` which is active-Low.
- AXI Streaming interface – `s_axis_aresetn/m_axis_aresetn` which is active-Low.
- Soft reset through the register interface

The AXI4-Lite interface reset and soft reset clears the entire design. AXI streaming interface reset clears the FIFO and the streaming interface logic. There is also a register bit to flush the internal FIFO alone.

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 Vivado 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) [\[Ref 3\]](#)
- *Vivado Design Suite User Guide: Getting Started* (UG910) [\[Ref 5\]](#)
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [\[Ref 6\]](#)

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 IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [\[Ref 2\]](#) for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values 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

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) [\[Ref 3\]](#) and the *Vivado Design Suite User Guide: Getting Started* (UG910) [\[Ref 5\]](#).

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

Figure 4-1 shows the Customize IP dialog box with information about customizing parameters in the Transmit Mode.

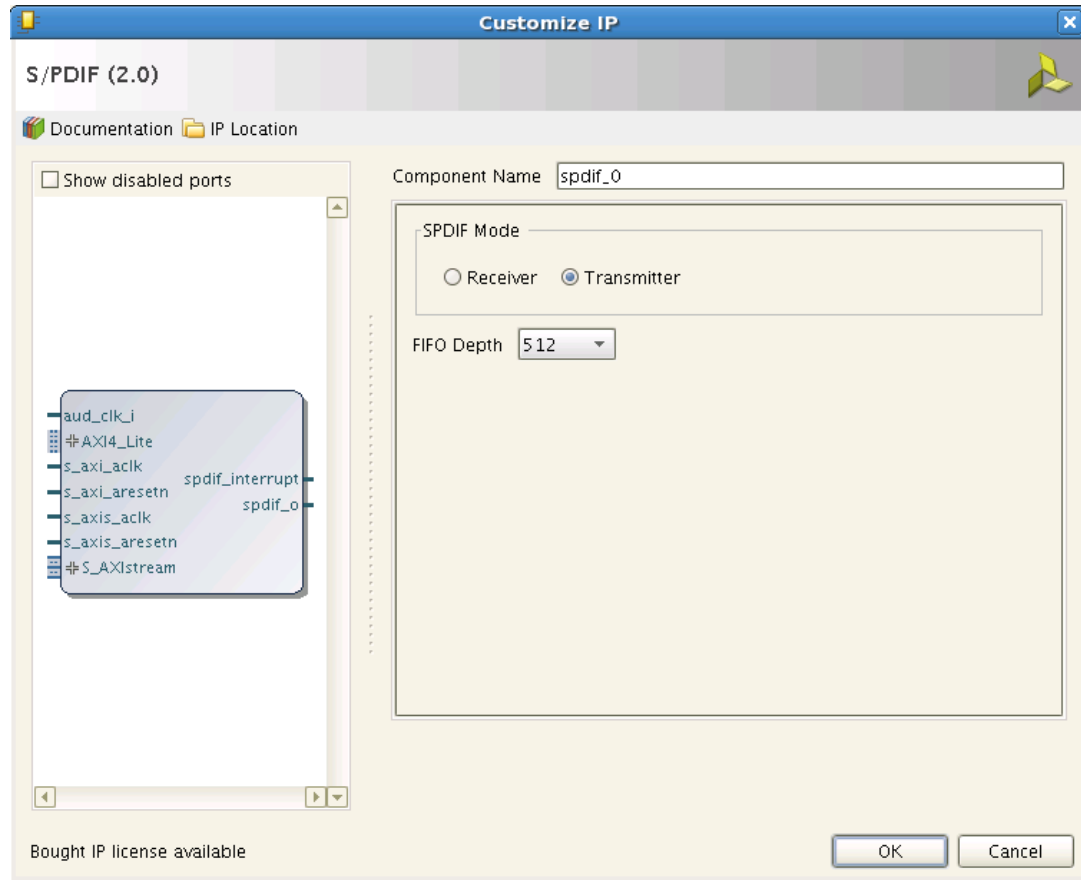


Figure 4-1: S/PDIF in Transmit Mode

- **Component Name** – Enter a name for the core instance. This example uses the name *spdif_0*.
- **S/PDIF Mode** – This selects the mode of operation of S/PDIF. It can either be a receiver or transmitter based on selection.
- **FIFO Depth** – This sets the Internal FIFO depth to store audio samples. Set the depth of FIFO based on the *aud_clk_i* and streaming clock connected (*m_axis_aclk* in case of S/PDIF Receiver/*s_axis_aclk* in case of S/PDIF Transmitter). This option needs to be set for both S/PDIF Receiver and Transmitter.

Figure 4-2 shows the Customize IP dialog box with information about customizing parameters in the Receive Mode.

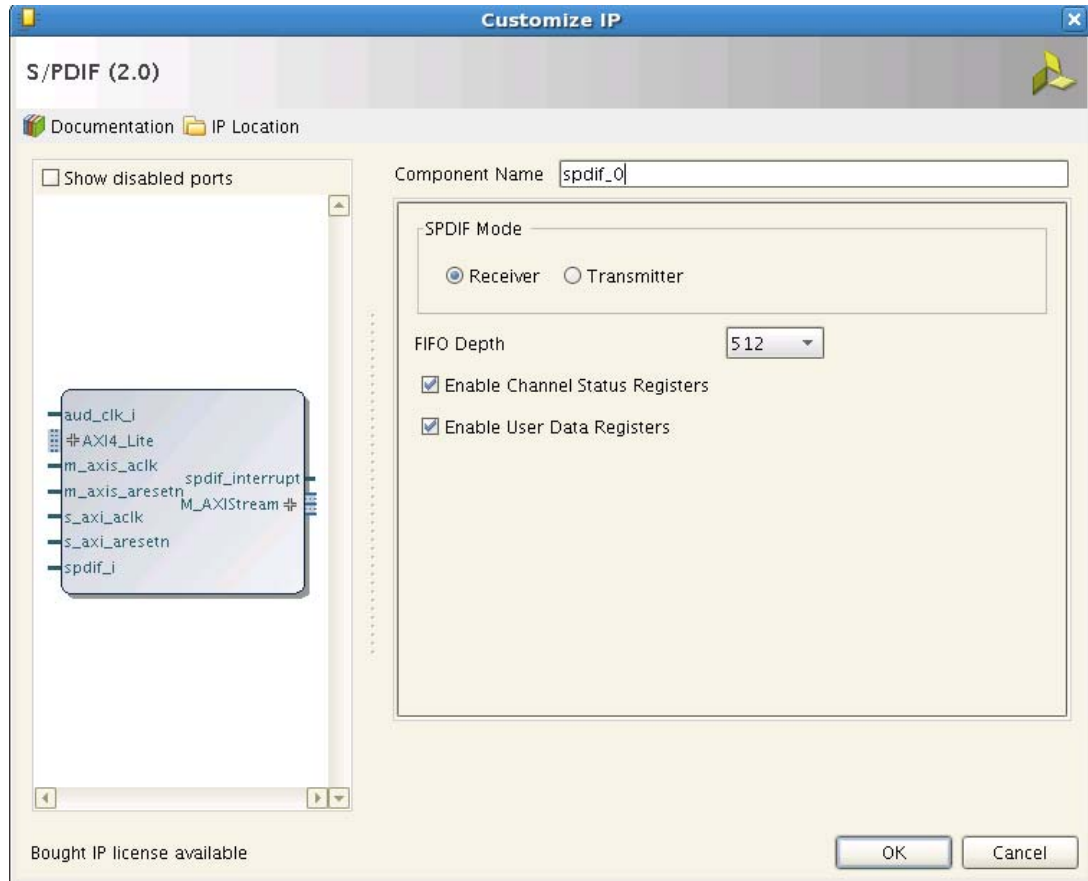


Figure 4-2: S/PDIF in Receive Mode

In addition to the stated S/PDIF Mode and FIFO Depth, select the following options for the S/PDIF in receive mode.

- **Enable Channel Status Registers** – Set this option when S/PDIF is selected in Receive mode. When set, S/PDIF Receiver captures the channel status data from the audio frames in the internal registers.
- **Enable User Data Registers** – Set this option when S/PDIF is selected in Receive mode. When set, S/PDIF Receiver captures the channel status data from the audio frames in the internal registers.

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 3].

Constraining the Core

This section defines the constraint requirements of the S/PDIF endpoint example design. An example Xilinx Design Constraints (XDC) is provided with the example design, which implements the constraints defined in this chapter.

When a Virtex[®]-7 FPGA is selected as the target device, the XDC is generated for an XC7VX485TFFG1157-1 device as an example. The example designs and XDCs can be retargeted for other devices.

Information provided in this section indicates which constraints to modify when targeting devices other than those shown in the example designs.

Required Constraints

This section is not applicable for this IP core.

Device, Package, and Speed Grade Selections

The S/PDIF cores can be implemented in the devices listed in the IP Facts table on [page 4](#) with the following attributes:

- Large enough to accommodate the cores
- With a fast enough speed grade to meet the frequency requirements

Clock Frequencies



IMPORTANT: *To operate the core at the highest performance rating, the following constraints must be present. Prorate these numbers if slower performance is desired.*

```
create_clock -name "TS_AUD_CLK_I" -period 10 -waveform {0 5.0} [get_ports AUD_CLK_I]
create_clock -name "TS_AXI_ACLK" -period 20 -waveform {0 10.0} [get_ports
S_AXI_ACLK]
```

When S/PDIF configured in Receive Mode:

```
create_clock -name "TS_M_AXIS_ACLK" -period 20 -waveform {0 10.0} [get_ports
M_AXIS_ACLK]
```

When S/PDIF configured in Transmit Mode:

```
create_clock -name "TS_S_AXIS_ACLK" -period 20 -waveform {0 10.0} [get_ports
S_AXIS_ACLK]
```

Clock Management

This section is not applicable for this IP core.

Clock Placement

This section is not applicable for this IP core.

Banking

This section is not applicable for this IP core.

Transceiver Placement

This section is not applicable for this IP core.

I/O Standard and Placement

This section is not applicable for this IP 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 6].

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 3].

Example Design

This chapter contains information about the example design provided in the Vivado® Design Suite.

Figure 5-1 and Figure 5-2 illustrate the S/PDIF receiver and transmitter example designs, respectively. The example designs consist of the following:

- S/PDIF design
- S/PDIF partner pair
- AXI4-Stream master model
- AXI4-Lite master model
- Stream checker
- HDL wrapper which instantiates all the blocks
- Design is synthesizable and tested with Xilinx® Vivado
- Customizable demonstration test bench to simulate the example design

The S/PDIF example designs have been tested with Vivado Design Suite and the Mentor Graphics Questa® SIM.

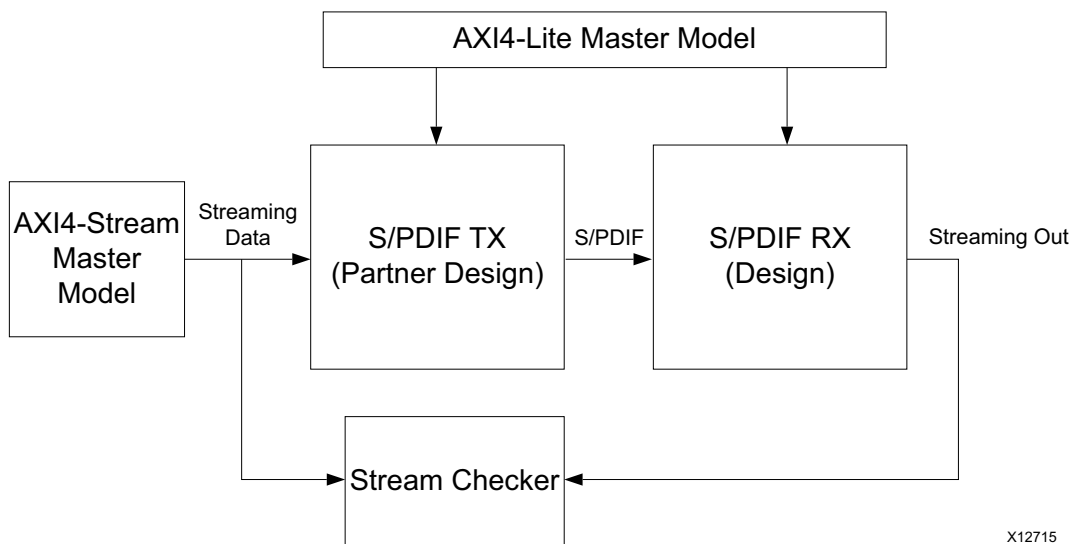


Figure 5-1: S/PDIF Receiver Example Design

X12715

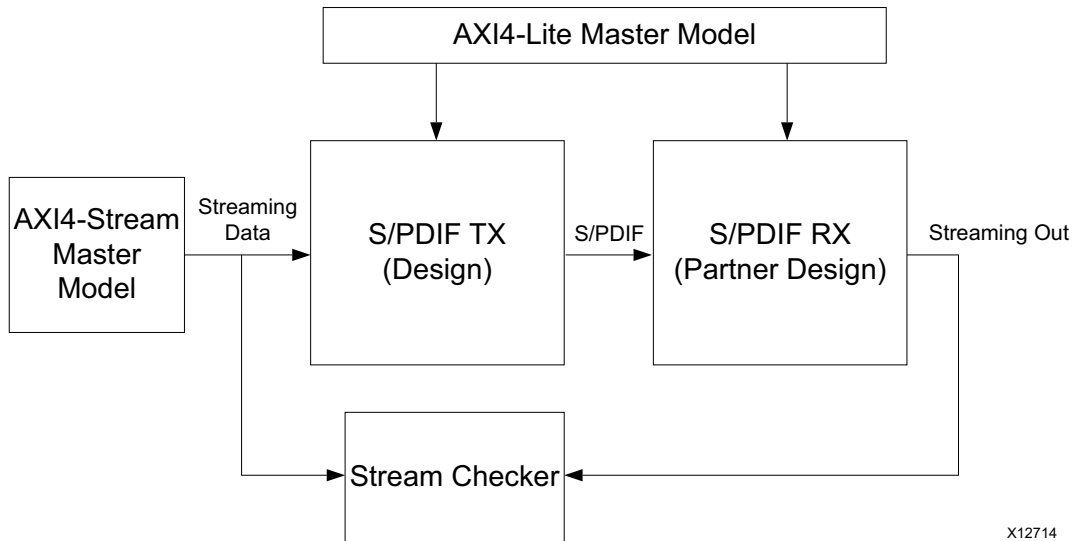


Figure 5-2: S/PDIF Transmitter Example Design

This section describes the functionality of the blocks used in the example design.

- **S/PDIF Partner Design** – Partner is the counterpart configuration of the DUT design. If DUT is configured in RX mode, the partner design is configured in TX mode and vice versa.
- **AXI4-Lite Master Model** – In the AXI4-Lite Master Model, the S/PDIF Control register is set to 32'h00000001 to enable the S/PDIF core in the RX mode. For S/PDIF TX, the partner Control register is set to 32'h0000000D to enable the S/PDIF TX mode.
- **AXI4-Stream Driver** – AXI4-Stream TX Mode driver generates 384 sub-frames of 32 bits to drive the AXI4-Stream slave interface of the S/PDIF core as defined in the AXI4-Stream protocol. This model generates the 4-bit preambles as required by the S/PDIF protocol. The 28-bit audio data generated by this model is an incremental pattern.
- **AXI4-Stream Checker** – This model compares the generated data from the S/PDIF RX with the AXI4-Stream data generated from the AXI4-Stream driver.

Test Bench

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

Test Bench Functionality

The test bench consists of the following:

- Test bench top file that instantiates the top-level example file and generates the necessary clocks required by the test bench.
- By reading the status signals from the stream checker (to verify the data on the S/PDIF output line) reports whether the data received correctly or any mismatch exists.

S/PDIF Core in RX Mode

This section describes the functionality of the blocks in the demonstration test bench in RX mode.

- **Clock Generator** – The sampling frequency at which the audio data is received on the core S/PDIF input line is fixed at 32 kHz. The signal, `aud_clk_i`, with 16.384 MHz frequency, is generated by the top-level module as required for the 32 kHz sampling frequency. The clock generator generates a reference clock for use by the test bench components.
- **AXI4-Lite Master Model** – In the AXI4-Lite Master Model, the S/PDIF control register is set to `32'h00000001` to enable the S/PDIF core in RX mode and S/PDIF TX partner control register is set to `32'h0000000D` to enable the S/PDIF TX mode.
- **AXI4-Stream Driver** – AXI4-Stream TX Mode driver generates 384 sub-frames of 32 bits to drive on the AXI4-Stream slave interface of the S/PDIF core as defined in the AXI4-Stream protocol. This model generates the 4-bit preambles as required by the S/PDIF protocol. The 28-bit audio data generated by this model is an incremental pattern.

The channel number is driven such that alternate channel IDs exist (channel 0 and channel 1).

- **AXI4-Stream Checker** – This model compares the generated data from S/PDIF core with the AXI4-Stream data generated from AXI4-Stream driver.

S/PDIF Core in TX Mode

This section describes the functionality of the blocks in the demonstration test bench in TX mode.

- **Clock Generator** – The sampling frequency at which the audio data is transmitted on the core S/PDIF output line is fixed at 32 kHz. The signal, `aud_clk_i` is generated by the top-level module as required for the 32 kHz sampling frequency. The signal, `aud_clk_i` is generated for 49.152 MHz.
- **AXI4-Lite Master Model** – In the AXI4-Lite Master Model, the S/PDIF control register is set to `32'h0000000D` to enable the S/PDIF core in TX mode. This configuration sets the TX clock configuration bits to a value of `4'h3` (divisor 24) and the S/PDIF RX partner control register is set to `32'h00000001` to enable the S/PDIF RX mode.
- **AXI4-Stream Driver** – The AXI4-Stream TX Mode Driver generates 1,384 sub-frames of 32 bits to drive on the AXI4-Stream slave interface of the S/PDIF core as defined in the AXI4-Stream protocol. This model generates the 4-bit preambles as required by the S/PDIF protocol. The 28-bit audio data generated by this model is an incremental pattern.

The channel number is driven such that alternate channel IDs exist (channel 0 and channel 1).

- **AXI4-Stream Checker** – This model compares the generated data from S/PDIF RX partner with the AXI4-Stream data generated from AXI4-Stream driver.

Migrating and Upgrading

This appendix contains information about migrating a design from ISE® to the Vivado® Design Suite, and for upgrading to a more recent version of the IP core. For customers upgrading in the Vivado Design Suite, important details (where applicable) about any port changes and other impact to user logic are included.

Migrating to the Vivado Design Suite

For information on migrating to the Vivado Design Suite, see the *ISE to Vivado Design Suite Migration Guide* (UG911) [Ref 7].

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.

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 S/PDIF, the [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 S/PDIF. 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 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 Records for the S/PDIF

AR: [54543](#)

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 S/PDIF 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 devices.

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 8].

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 is 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.

Many of these common issues can also be applied to debugging design simulations. Details are provided in the General Checks section.

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.

- Check all the clocks are toggling or not.
 - Check the active state of each reset and make sure no interface is in reset condition.
 - Check the `aud_clk_i` requirements are met for S/PDIF Transmitter/S/PDIF Receiver.
 - Check whether the core is configured and then enabled or not before looking at the functionality of the core.
-

Interface Debug

AXI4-Lite Interfaces

Write a specific value to any read/write register and it reads back the same register 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 `aud_clk_i` inputs are connected and toggling.
- The interface is not being held in reset, and `s_axi_aresetn` is an active-Low reset.
- If the simulation has been run, verify in simulation and/or 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 `s_axis_aclk` input for S/PDIF Transmitter and `m_axis_aclk` input for S/PDIF receiver are connected and toggling.
- Check that the AXI4-Stream waveforms are being followed.
- Check core configuration.

Other Interfaces

In S/PDIF Receiver mode:

- Check if the S/PDIF interface line is active or not in case of S/PDIF Receiver.
- Check the `aud_clk_i` requirement (that is, the clock should be minimum of eight times the S/PDIF bit rate).
- Check the Status register whether the core capturing the bit rate as expected or not.

In S/PDIF Transmitter mode:

- Check the `aud_clk_i` frequency. It should be harmonic to the Audio sampling rate.
- Check whether the proper divisor value is set in Control register to generate a S/PDIF line rate properly.

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. [AMBA AXI4-Stream Protocol Specification](#)
2. *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator* ([UG994](#))
3. *Vivado® Design Suite User Guide: Designing with IP* ([UG896](#))
4. *LogiCORE™ IP Interrupt Control Data Sheet* ([DS516](#))
5. *Vivado Design Suite User Guide: Getting Started* ([UG910](#))
6. *Vivado Design Suite User Guide: Logic Simulation* ([UG900](#))
7. *ISE® to Vivado Design Suite Migration Guide* ([UG911](#))
8. *Vivado Design Suite User Guide: Programming and Debugging* ([UG908](#))
9. *Synthesis and Simulation Guide* ([UG626](#))
10. *IEC-60958-3 Standard Specification*

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
11/18/2015	2.0	Added support for UltraScale+ families.
04/02/2014	2.0	<ul style="list-style-type: none"> • Updated Fig. 1-1 S/PDIF Block Diagram. • Added Applications section. • Added Performance section. • Updated description in Resource Utilization section. • Updated Example Design and Test Bench chapters.
03/20/2013	4.0	<ul style="list-style-type: none"> • Updated to core version 2.0 and Vivado Design Suite only. • Updated to Questa SIM. • Updated Resource Utilization tables. • Updated Reset descriptions. • Updated Table 3-4 Design Parameters and Table 3-5 Parameter I/O. • Added Fig. 4-1 and 4-2 GUI. • Updated Output Generation section. • Updated code in Clock Frequencies section. • Removed Verification appendix. • Updated Appendix Debug section.
12/18/2013	2.0	<ul style="list-style-type: none"> • Revision number reverted to 2.0 to align with core version number 2.0. • Added IP Integrator and UltraScale support. • Added Simulation, Synthesis, Example Design, and Test Bench chapters. • Updated Migrating chapter and Debugging Appendix.
12/18/2012	3.0	<ul style="list-style-type: none"> • Updated to ISE Design Suite 14.4 and Vivado Design Suite 2012.4. • Updated description to S/PDIF Control register Bits[5:2]. • Updated Appendix Debug section.
07/25/2012	2.0	Updated for Vivado Design Suite 2012.2, Zynq features, and ISE 14.2.
04/24/2012	1.0	Initial Xilinx release

Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at <http://www.xilinx.com/legal.htm#tos>.

© Copyright 2012–2015 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. AMBA, AMBA Designer, ARM, ARM1176JZ-S, CoreSight, Cortex, and PrimeCell are trademarks of ARM in the EU and other countries. All other trademarks are the property of their respective owners.