

# SPDIF/AES3 v2.0

## *LogiCORE IP Product Guide*

**Vivado Design Suite**

**PG045 April 4, 2018**

# Table of Contents

## IP Facts

### Chapter 1: Overview

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

### Chapter 2: Product Specification

|                         |    |
|-------------------------|----|
| Standards .....         | 12 |
| Performance .....       | 12 |
| Port Descriptions ..... | 13 |
| Register Space .....    | 16 |

### Chapter 3: Designing with the Core

|  |    |
|--|----|
| Clocking .....                                   | 24 |
| Resets .....                                     | 25 |
| Electrical Circuit Considerations on Board ..... | 25 |

### 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 |
| SPDIF/AES3 Core in RX Mode ..... | 34 |
| SPDIF/AES3 Core in TX Mode ..... | 35 |

## Appendix A: 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 |
| Documentation Navigator and Design Hubs ..... | 42 |
| References .....                              | 42 |
| Revision History .....                        | 44 |
| Please Read: Important Legal Notices .....    | 45 |

## Introduction

The SPDIF/AES3 core is a digital audio interface controller that implements the International Electronic Commission (IEC) 60958-3 interface for transmitting and receiving audio data.

AES3 (also known as AES/EBU) is an audio interface standard for the exchange of digital audio signals between professional audio and video devices. AES3 was jointly developed by Audio Engineering Society (AES) and the European Broadcasting Union (EBU). An AES3 signal can carry two channels of PCM audio. AES3 is available in a consumer-grade variant known as S/PDIF. AES3 is commonly used in professional video and audio equipment used in broadcast studios such as cameras, servers, and switches. S/PDIF is more common in the consumer version of the audio equipment.

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 SPDIF/AES3 core is stored in the core internal FIFO, allowing the system to process a relatively slow audio stream.

## Features

- Configurable as an SPDIF/AES3 audio data transmitter or an SPDIF/AES3 audio data receiver
- Configurable FIFO buffer stores the audio sample data
- IEC 60958-3 standard SPDIF/AES3 digital audio bus interface
- Two audio channels
- Audio sample lengths of 16/20/24 bits
- Variable sampling rates (32/44.1/48/88.2/96/176.4/192 kHz)
- The transmitter sends the invalid null audio frames over the SPDIF/AES3 line in case of a FIFO under-run condition

| LogiCORE IP Facts Table   |   |
|---|---|
| Core Specifics  |   |
| Supported Device Family <sup>(1)</sup>                            | UltraScale+™ Families, UltraScale™ Architecture, Zynq®-7000 All Programmable SoC, Xilinx 7 series |
| Supported User Interfaces   | SPDIF/AES3, AXI4-Stream, AXI4-Lite  |
| Resources   | Performance and Resource Utilization web page   |
| Provided with Core  |   |
| Design Files  | Encrypted HDL   |
| Example Design  | Verilog, VHDL   |
| Test Bench  | VHDL  |
| Constraints File  | XDC   |
| Simulation Model  | Verilog and VHDL Structural Models  |
| Supported S/W Driver  | Provided  |
| Tested Design Flows <sup>(2)</sup>                                |   |
| Design Entry  | Vivado® Design Suite<br>Vivado  |
| Simulation  | For supported simulators, see the <a href="#">Xilinx Design Tools: Release Notes Guide</a> .      |
| Synthesis   | Vivado Synthesis  |
| Support   |   |
| Provided by Xilinx at the <a href="#">Xilinx Support web page</a> |   |

### 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 SPDIF/AES3 core is compatible with the SPDIF or AES3 protocol. It can be used in a receive or transmit mode and delivers or accepts audio data from an AXI4-Stream input. The SPDIF/AES3 core is designed for use in audio systems, and provide AES3 interface for LogiCORE Ips such as UHD-SDI, Display port and HDMI core for audio data transfers. [Figure 1-1](#) shows the SPDIF/AES3 block diagram.

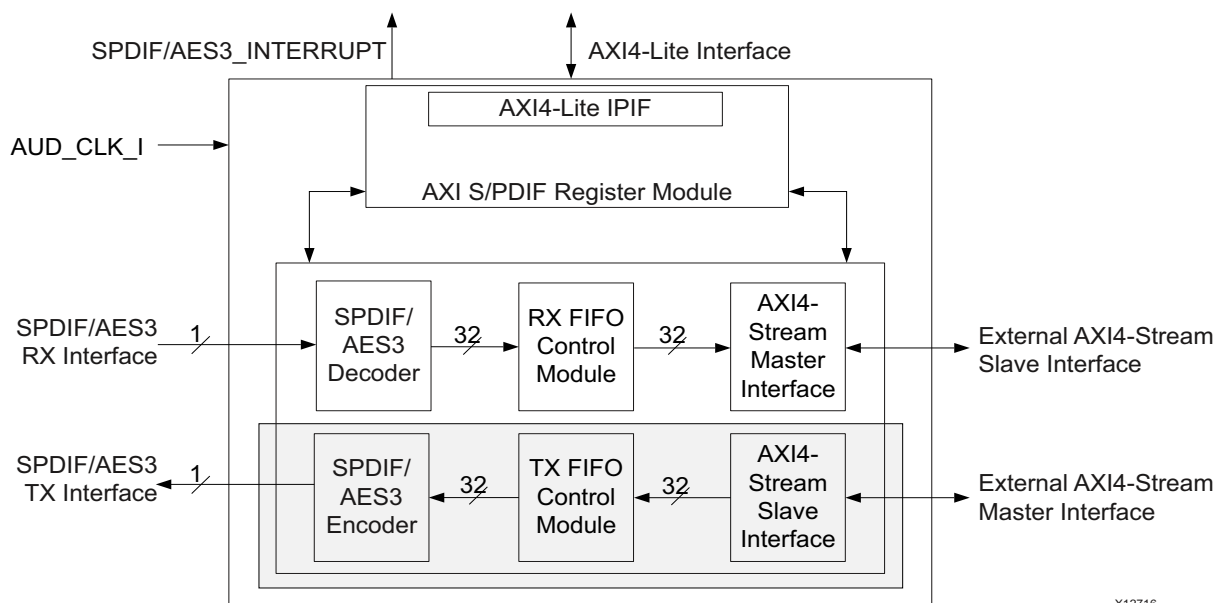


Figure 1-1: SPDIF/AES3 Block Diagram

The SPDIF/AES3 core can operate in two modes:

- **SPDIF/AES3 Receiver** – Receives SPDIF/AES3 audio data and sends it through the AXI4-Stream interface
- **SPDIF/AES3 Transmitter** – Receives audio data through the AXI4-Stream interface and transmits it through the SPDIF/AES3 interface

The core contains the following blocks in SPDIF/AES3 receive mode:

- AXI4-Lite IPIF
- AXI SPDIF/AES3 Register Module
- SPDIF/AES3 Decoder
- RX FIFO Control Module
- AXI4-Stream Master Interface

The core contains the following blocks in SPDIF/AES3 transmit mode:

- AXI4-Lite IPIF
- AXI SPDIF/AES3 Register Module
- AXI4-Stream Slave Interface
- TX FIFO Control Module
- SPDIF/AES3 Encoder

## SPDIF/AES3 Register Module

This section describes the blocks within the SPDIF/AES3 register module.

### ***AXI4-Lite IPIF***

The SPDIF/AES3 register module provides the read/write control logic for the SPDIF/AES3 core register set. The registers are accessible by the AXI4-Lite master interface through the AXI4-Lite IPIF module, which is part of the SPDIF/AES3 register module.

The data width of the AXI4-Lite interface is fixed at 32 bits. The registers are defined in [Table 2-3, page 16](#).

The interrupt control and soft reset functionality are also implemented as part of the SPDIF/AES3 register module. The SPDIF/AES3 core can be reset by writing 0xA to the soft reset register. For an SPDIF/AES3 transmitter, an SPDIF/AES3 interrupt can be generated based on the FIFO Full/FIFO empty conditions. For an SPDIF/AES3 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 SPDIF/AES3 line or if the SPDIF/AES3 core receives the start of block over the SPDIF/AES3 line. In the Rx mode, the interrupt is also triggered when the channel-status value changes from one block to another.

## SPDIF/AES3 Decoder

The enable bit in the Control register of the SPDIF/AES3 register module has to be set to enable the SPDIF/AES3 decoder module. The SPDIF/AES3 decoder recovers data from the bi-phase mark coded SPDIF/AES3 data stream (see [Bi-Phase Mark Code](#)). The audio clock

frequency should be at least sixteen 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  $16 \times 64 \times 192 \text{ kHz} = 98.304 \text{ MHz}$  to recover the data samples. As per the SPDIF/AES3 Protocol Preamble violates the bi-phase mark code format, SPDIF/AES3 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 SPDIF/AES3 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 SPDIF/AES3 register module.

This module detects the BMC/Preamble errors over the SPDIF/AES3 line and reports to the SPDIF/AES3 register module. This module also generates a recovered clock that is equivalent to the Fs of the incoming line.

## RX FIFO Control Module

The Asynchronous RX FIFO is used to store the 32-bit audio data received from the SPDIF/AES3 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 SPDIF/AES3 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 SPDIF/AES3 register module. Similarly, the RX FIFO empty interrupt and corresponding status are updated through the SPDIF/AES3 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, channelA audio data, and channelB audio data are listed in [Table 1-2](#).

Table 1-2: Preamble

| Bits[3:0] | Description                                   |
|-----------|---|
| 0001      | Start of audio block and Channel A audio data |
| 0010      | ChannelA audio data                           |
| 0011      | ChannelB 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 SPDIF/AES3 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 SPDIF/AES3 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 SPDIF/AES3 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 SPDIF/AES3 register module. This condition can occur when the SPDIF/AES3 enable bit is not set in the Control register of the SPDIF/AES3 register module, and the target is sending continuous streaming data or when the FIFO Size is not sufficient. After a reset, the IP starts transmitting data (when enabled) only when the FIFO is at least half filled.



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 SPDIF/AES3 transmitter to send over the SPDIF/AES3 line.

## SPDIF/AES3 Encoder

This module has to get the clock configuration bits through the Control register to know the bit rate before SPDIF/AES3 data transmission starts. The enable bit in the Control register of the SPDIF/AES3 registers module must be set to enable the SPDIF/AES3 encoder module. The SPDIF/AES3 encoder converts the 32-bit parallel data received from the TX FIFO to serial data. The serial data is transferred over the SPDIF/AES3 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).

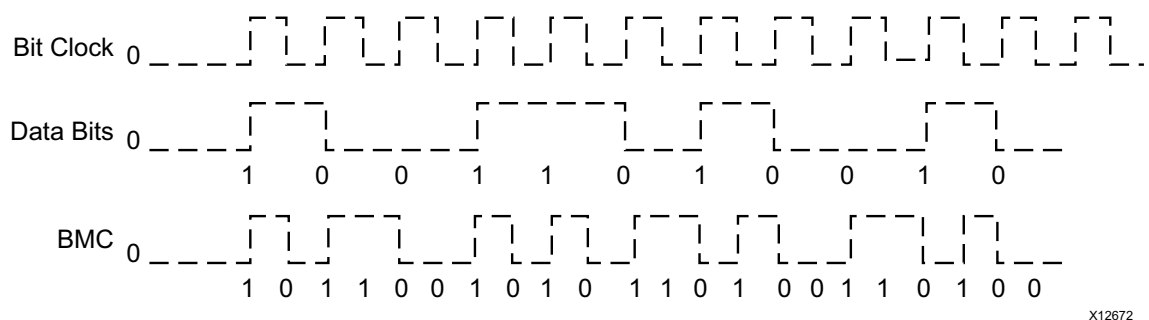
In case of a TX FIFO under-run condition, this module sends the null audio frames over the SPDIF/AES3 line with the validity bit set. When the validity bit is set to 1, it means per the SPDIF/AES3 protocol that the audio sample is invalid and the codec has to ignore the sample.



**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 SPDIF/AES3 interface (IEC-60958) is a consumer version of the AES/EBU-interface. The SPDIF/AES3 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.



X12672

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, SPDIF/AES3 audio data is transferred in the BMC format. Preambles violate the BMC to identify the channel information by the SPDIF/AES3 receivers.

---

## Feature Summary

The LogiCORE IP SPDIF/AES3 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 SPDIF/AES3 audio data transmitter or an SPDIF/AES3 audio data receiver. The IEC 60958-3 standard SPDIF/AES3 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<sup>®</sup> 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 SPDIF/AES3 core is stored in the core internal FIFO, allowing the system to process a relatively slow audio stream.

---

## Applications

The SPDIF/AES3 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.

**Note:** The SPDIF/AES3 core can be used to transmit and receive SPDIF as well as AES3 data. A necessary circuit (electrical interface, impedance matching) should be in place, on the board, based on the usage.

---

## Unsupported Features

The SPDIF/AES3 core does not support non-linear PCM encoded audio data streams. The core does not verify the Parity or CRC.

---

## Licensing and Ordering

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 SPDIF/AES3 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 <sub>Max</sub> (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         |

## Port Descriptions

The SPDIF/AES3 Input/Output (I/O) signals are listed and described in [Table 2-2](#).

Table 2-2: I/O Signal Description

| Signal Name                                     | Interface | I/O | Initial State | Description   |
|---|-----------|-----|---------------|---|
| <b>System Signals</b>                           |           |     |               |   |
| aud_clk_i                                       | System    | I   | –             | Audio clock input used at the SPDIF/AES3 interface  |
| spdif_interrupt                                 | System    | O   | 0             | SPDIF/AES3 core interrupt output. When the interrupt occurs, this signal is continuously 1 until cleared/disabled.      |
| <b>AXI4-Lite Interface System Signals</b>       |           |     |               |   |
| s_axi_aclk                                      | System    | I   | –             | AXI4-Lite clock   |
| s_axi_aresetn                                   | System    | I   | –             | AXI4-Lite reset, active-Low   |
| <b>AXI4-Lite Write Address Channel Signals</b>  |           |     |               |   |
| 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. |
| <b>AXI4-Lite Write Data Channel Signals</b>     |           |     |               |   |
| 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 SPDIF/AES3 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.  |
| <b>AXI4-Lite Write Response Channel Signals</b> |           |     |               |   |
| 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 2-2: I/O Signal Description (Cont'd)

| Signal Name                                   | Interface    | I/O | Initial State | Description   |
|---|--------------|-----|---------------|---|
| <b>AXI4-Lite Read Address Channel Signals</b> |              |     |               |   |
| 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.  |
| <b>AXI4-Lite Read Data Channel Signals</b>    |              |     |               |   |
| 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.  |
| <b>SPDIF/AES3 RX Interface Signals</b>        |              |     |               |   |
| spdif_i                                       | SPDIF/AES3   | I   | –             | Audio input from the SPDIF/AES3 interface   |
| rxclk_out                                     | NA           | O   | 0             | Recovered Fs signal based on the incoming SPDIF/AES3 input  |
| <b>SPDIF/AES3 TX Interface Signals</b>        |              |     |               |   |
| spdif_o                                       | SPDIF/AES3   | O   | 0             | Audio output to the SPDIF/AES3 interface  |
| <b>AXI4-Stream Master Interface Signals</b>   |              |     |               |   |
| 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.<br>0 = Write data is not valid<br>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.<br>0 = Not ready to receive data<br>1 = Ready to receive data               |

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

| Signal Name                                | Interface    | I/O | Initial State | Description   |
|--|--------------|-----|---------------|---|
| m_axis_tdata[c_axis_tdata_width – 1:0]     | M_AXISStream | O   | 0             | AXI4-Stream Data Out  |
| 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.   |
| <b>AXI4-Stream Slave Interface Signals</b> |              |     |               |   |
| 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.<br>0 = Write data is not valid<br>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.<br>0 = Not ready to receive data<br>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 2-3 specifies the offset address, register name, and accessibility of each firmware addressable register from the three classes of registers within the SPDIF/AES3 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 SPDIF/AES3 receiver logic. Channel Status registers hold the 192-bit channel status information received over the SPDIF/AES3 input when the SPDIF/AES3 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 SPDIF/AES3 user data bits into user data registers is configurable. When the User Enabled registers are enabled, only these registers are part of the SPDIF/AES3 receiver logic. User Data registers hold the 192-bit user data received over the SPDIF/AES3 input when the SPDIF/AES3 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 2-3: Register Map

| Offset Address (Hex)   | Register Name                   | Description                              |
|--|---------------------------------|--|
| <b>Interrupt Registers</b>                                   |                                 |  |
| 0x1C   | Global Interrupt Enable (GIE)   | Device Global interrupt enable register  |
| 0x20   | Interrupt Status Register (ISR) | IP interrupt status register             |
| 0x28   | Interrupt Enable Register (IER) | IP interrupt enable register             |
| <b>Soft Reset Register</b>                                   |                                 |  |
| 0x40   | Soft Reset Register             | Soft Reset Register                      |
| <b>SPDIF/AES3 Configuration, Control, and Data Registers</b> |                                 |  |
| 0x44   | SPDIF/AES3 Control Register     | Control register                         |
| 0x48   | SPDIF/AES3 Status Register      | Status register                          |
| 0x4C   | Channel Status Register0        | Audio Channel status bits 31 downto 0    |
| 0x50   | Channel Status Register1        | Audio Channel status bits 63 downto 32   |
| 0x54   | Channel Status Register2        | Audio Channel status bits 95 downto 64   |
| 0x58   | Channel Status Register3        | Audio Channel status bits 127 downto 96  |
| 0x5C   | Channel Status Register4        | Audio Channel status bits 159 downto 128 |



Table 2-3: Register Map (Cont'd)

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

**Notes:**

1. The soft reset functionality is implemented by the soft\_reset module.

## Global Interrupt Enable (GIE)

The Global Interrupt Enable register, described in [Table 2-4](#), has a single defined bit that globally enables the final interrupt out to the system.

Table 2-4: Global Interrupt Enable Register (Offset 0x1C)

| Bits | Name   | Reset Value | Access Type | Description   |
|------|--------|-------------|-------------|---|
| 31   | GIE    | 0           | R/W         | Global Interrupt Enable<br>0 = All interrupts disabled. No interrupts from SPDIF/AES3<br>1 = Unmasked SPDIF/AES3 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 SPDIF/AES3 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 2-5](#).

**Table 2-5: Interrupt Status Register (Offset 0x20)**

| Bits | Name                  | Reset Value | Access Type              | Description  |
|------|-----------------------|-------------|--------------------------|--|
| 31:6 | Unused                | 0           | N/A                      | Reserved   |
| 5    | Channel Status Change | 0           | Read/Toggle on writing 1 | This bit is set whenever a change in the channel status information is detected.   |
| 4    | Preamble Error        | 0           | Read/Toggle on writing 1 | This bit is set when the incorrect preamble format is received over the SPDIF/AES3 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 SPDIF/AES3 audio data bits in receive mode (except for the preamble). The core drops the affected sample in case of a BMC error. |
| 2    | Start Of Block        | 0           | Read/Toggle on writing 1 | This bit is set when the SPDIF/AES3 core is in receive mode and when it detects the start of block preamble over the S/PDIF_I input. This bit is set from the second block onwards.                          |
| 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 SPDIF/AES3 interrupts. The Interrupt Enable register bit fields are described in [Table 2-6](#).

Table 2-6: Interrupt Enable Register (Offset 0x28)

| Bits | Name                              | Reset Value | Access Type | Description  |
|------|-----------------------------------|-------------|-------------|--|
| 31:6 | Unused                            | N/A         | N/A         | Reserved   |
| 5    | Channel Status Change             | 0           | R/W         | This bit must be set to generate the Channel Status change interrupt. This is applicable only in receive mode.   |
| 4    | Preamble Error                    | 0           | R/W         | This bit must be set to generate the preamble error interrupt. In transmit mode, this bit is unused.   |
| 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 SPDIF/AES3 core is in transmit mode and the same bit must be set when the SPDIF/AES3 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 SPDIF/AES3 core is in transmit mode and the same bit has to be set when the SPDIF/AES3 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 SPDIF/AES3 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 SPDIF/AES3 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 2-7](#).

Table 2-7: 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 SPDIF/AES3 controller. Writing any other value results in an AXI4 transaction acknowledgment with SLVERR and no reset occurs. |

## SPDIF/AES3 Control Register

The SPDIF/AES3 Control register is a read and write register that configures the SPDIF/AES3 core. This register has an SPDIF/AES3 enable bit, a TX/RX FIFO flush bit, and clock configuration bits. The SPDIF/AES3 Control register bit fields are described in [Table 2-8](#).

**Table 2-8: SPDIF/AES3 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 SPDIF/AES3 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.<br>Bits Division Number:<br>0000 = 4<br>0001 = 8<br>0010 = 16<br>0011 = 24<br>0100 = 32<br>0101 = 48<br>0110 = 64<br>Others = Reserved |
| 1    | 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    | TX/RX Enable                | 0           | R/W         | This bit must be set to 1 to enable the SPDIF/AES3 core.   |

## SPDIF/AES3 Status Register

The SPDIF/AES3 Status register is a read-only register that contains the status of the SPDIF/AES3 core. The SPDIF/AES3 Status register bit fields are described in [Table 2-9](#).

Table 2-9: SPDIF/AES3 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           | <p>These bits are updated with the number of audio clocks for the SPDIF/AES3 data bit period. This audio clock count is recovered by the SPDIF/AES3 decoder module. This count gives the approximate count for the SPDIF/AES3 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. This value can be used to determine the appx Fs of the incoming SPDIF/AES3 frame as follows:</p> $F_s = 1000000 / (\text{Sample\_clock\_count} * 64 * \text{CLK\_PERIOD})$ <p>where, CLK_PERIOD is the period (in ns) of the clock connected to aud_clk_i.</p> |

## Channel Status Registers

A set of six configurable registers store the 192-bit SPDIF/AES3 Audio Channel Status information. These registers are active when the SPDIF/AES3 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 SPDIF/AES3. The Channel Status register bit fields are described in [Table 2-10](#). For complete descriptions of these bit fields, see the IEC-60958-3 specification.

Table 2-10: 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 31 downto 0 of the audio channel status information received over SPDIF/AES3.    |
| 31:0 | Channel Status Register1 | 0           | R           | This register holds bits 63 downto 32 of the audio channel status information received over SPDIF/AES3.   |
| 31:0 | Channel Status Register2 | 0           | R           | This register holds bits 95 downto 64 of the audio channel status information received over SPDIF/AES3.   |
| 31:0 | Channel Status Register3 | 0           | R           | This register holds bits 127 downto 96 of the audio channel status information received over SPDIF/AES3.  |
| 31:0 | Channel Status Register4 | 0           | R           | This register holds bits 159 downto 128 of the audio channel status information received over SPDIF/AES3. |
| 31:0 | Channel Status Register5 | 0           | R           | This register holds bits 191 downto 160 of the audio channel status information received over SPDIF/AES3. |

## Channel A User Data Registers

A set of six configurable registers store the 192-bit SPDIF/AES3 Channela User data information. These registers are active when the SPDIF/AES3 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 2-11](#).

**Table 2-11: 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 31 downto 0 of the user data information received over SPDIF/AES3 channel a.        |
| 31:0 | Channela User Data Register1 | 0           | R           | This register holds bits 63 downto 32 of the audio user data information received over SPDIF/AES3 channel a. |
| 31:0 | Channela User Data Register2 | 0           | R           | This register holds bits 95 downto 64 of the user data information received over SPDIF/AES3 channel a.       |
| 31:0 | Channela User Data Register3 | 0           | R           | This register holds bits 127 downto 96 of the user data information received over SPDIF/AES3 channel a.      |
| 31:0 | Channela User Data Register4 | 0           | R           | This register holds bits 159 downto 128 of the user data information received over SPDIF/AES3 channel a.     |
| 31:0 | Channela User Data Register5 | 0           | R           | This register holds bits 191 downto 160of the user data information received over SPDIF/AES3 channel a.      |

## Channel B User Data Registers

A set of six configurable registers store the 192-bit SPDIF/AES3 Channelb User data information. These registers are active when the SPDIF/AES3 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 2-12](#).

**Table 2-12: 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 31 downto 0 of the user data information received over SPDIF/AES3 channel b.        |
| 31:0 | Channelb User Data Register1 | 0           | R           | This register holds bits 63 downto 32 of the audio user data information received over SPDIF/AES3 channel b. |

**Table 2-12: Channel B User Data Registers (Offsets 0x7C to 0x90) (Cont'd)**

| Bits | Name                         | Reset Value | Access Type | Description  |
|------|------------------------------|-------------|-------------|--|
| 31:0 | Channelb User Data Register2 | 0           | R           | This register holds bits 95 downto 64 of the user data information received over SPDIF/AES3 channel b.   |
| 31:0 | Channelb User Data Register3 | 0           | R           | This register holds bits 127 downto 96 of the user data information received over SPDIF/AES3 channel b.  |
| 31:0 | Channelb User Data Register4 | 0           | R           | This register holds bits 159 downto 128 of the user data information received over SPDIF/AES3 channel b. |
| 31:0 | Channelb User Data Register5 | 0           | R           | This register holds bits 191 downto 160 of the user data information received over SPDIF/AES3 channel b. |

# Designing with the Core

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

---

## Clocking

The SPDIF/AES3 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 SPDIF/AES3 audio interface works with this clock. The frequency of the clock should be  $\geq 1024$  times the audio sampling rate for SPDIF/AES3 in receive mode. In receive mode, if the incoming  $F_s$  is not known, then it is recommended to connect a 200 MHz clock to this port.

For SPDIF/AES3 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 SPDIF/AES3 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 SPDIF/AES3 register access works with this clock.
- **s\_axis\_aclk/m\_axis\_aclk** – This is the audio streaming interface clock.



## Resets

The SPDIF/AES3 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.

## Electrical Circuit Considerations on Board

You must ensure that a proper electrical circuit is present on the board for SPDIF or AES3 connectivity. For more details, see the AES3 specifications ([www.aes.org](http://www.aes.org)).

The Xilinx reference board supports unbalanced 75  $\Omega$  coaxial connections that are widely used in the broadcast industry to support the AES3 audio transmission. The AES3 input and output interfaces support unbalanced HD-BNC 75  $\Omega$  coaxial connections per AES3-4-2009, Annex D.

The AES3 input circuit utilizes an audio transformer circuit which provides a 75  $\Omega$  impedance unbalanced input. The Transformer magnetics convert the unbalanced signal to balanced (differential 100  $\Omega$ ) signal that sources a differential to single ended converter, which in turn drives the singled ended input to the Xilinx FPGA.

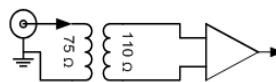


Figure 3-1: Simplified Input Circuit Diagram

On the AES3 output side, the FPGAs output is a 1.2V LVCMOS complementary pair that drives the 110  $\Omega$  balanced side of the audio transformer, which is converted to an unbalanced 75  $\Omega$  singled ended output signal to the 75  $\Omega$  coaxial HD-BNC connector.



Figure 3-2: Simplified Output Circuit Diagram

Since the IP core supports up to 192 KHz stereo audio, the AES3 input and output circuitry is designed to handle a maximum data bit rate of  $192\text{KHz} \times 2\text{ch} \times 32\text{bit/ch} = 12.288\text{ Mbps}$ . The AES3 electrical interface specifications are defined in AES3-4-2009, Annex D.

# 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) [\[Ref 4\]](#)
- *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 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) [\[Ref 4\]](#).

**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.

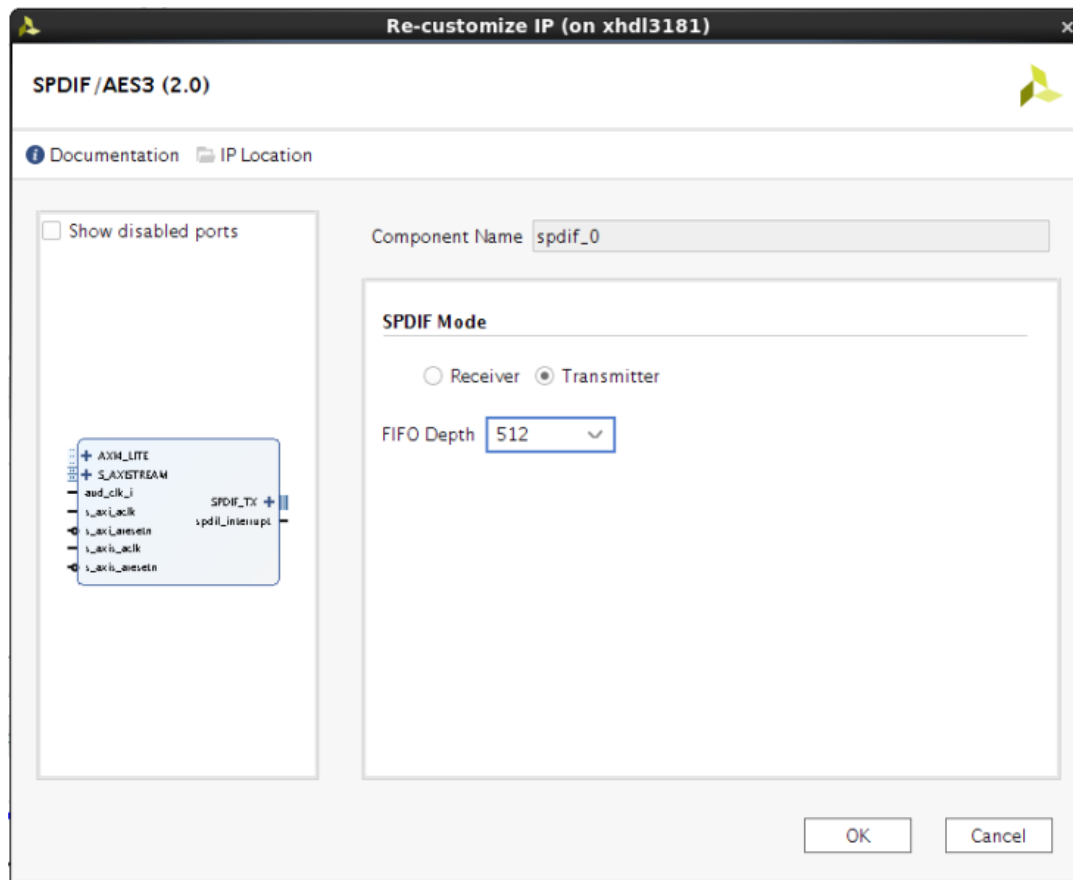


Figure 4-1: SPDIF/AES3 in Transmit Mode

- **Component Name** – Enter a name for the core instance. This example uses the name *spdif\_0*.
- **SPDIF/AES3 Mode** – This selects the mode of operation of SPDIF/AES3. 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 SPDIF/AES3 Receiver/*s\_axis\_aclk* in case of SPDIF/AES3 Transmitter). This option needs to be set for both SPDIF/AES3 Receiver and Transmitter. The actual depth of the FIFO is one less than what is configured.

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

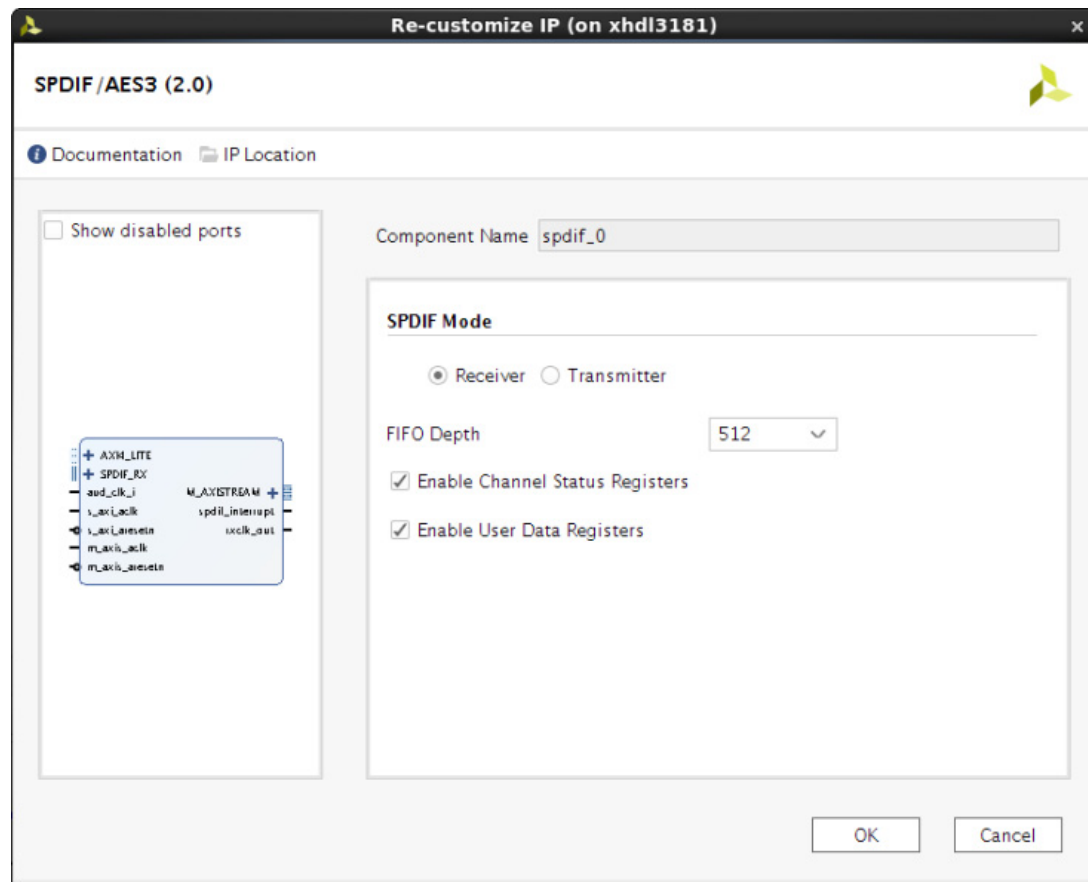


Figure 4-2: SPDIF/AES3 in Receive Mode

In addition to the stated SPDIF/AES3 Mode and FIFO Depth, select the following options for the SPDIF/AES3 in receive mode.

- **Enable Channel Status Registers** – Set this option when SPDIF/AES3 is selected in Receive mode. When set, SPDIF/AES3 Receiver captures the channel status data from the audio frames in the internal registers.
- **Enable User Data Registers** – Set this option when SPDIF/AES3 is selected in Receive mode. When set, SPDIF/AES3 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 SPDIF/AES3 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<sup>®</sup>-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 SPDIF/AES3 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]
```

### 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) [\[Ref 5\]](#).

---

## 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. This example design cannot be taken to any board and is primarily meant to understand the IP behavior.

Figure 5-1 and Figure 5-2 illustrate the SPDIF/AES3 receiver and transmitter example designs, respectively. The example designs consist of the following:

- SPDIF/AES3 design
- SPDIF/AES3 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 SPDIF/AES3 example designs have been tested with Vivado Design Suite and the Mentor Graphics Questa® SIM.

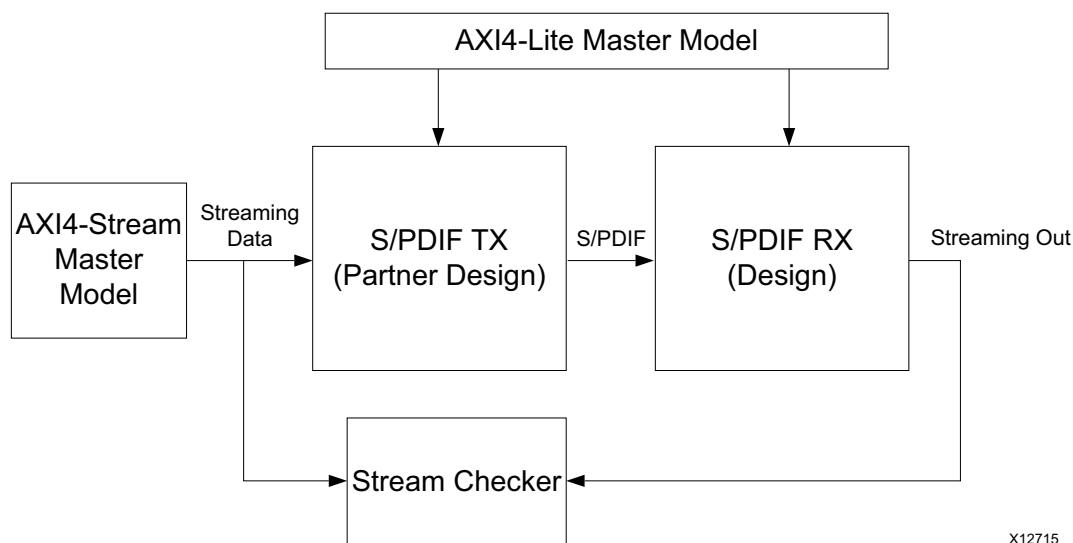


Figure 5-1: SPDIF/AES3 Receiver Example Design



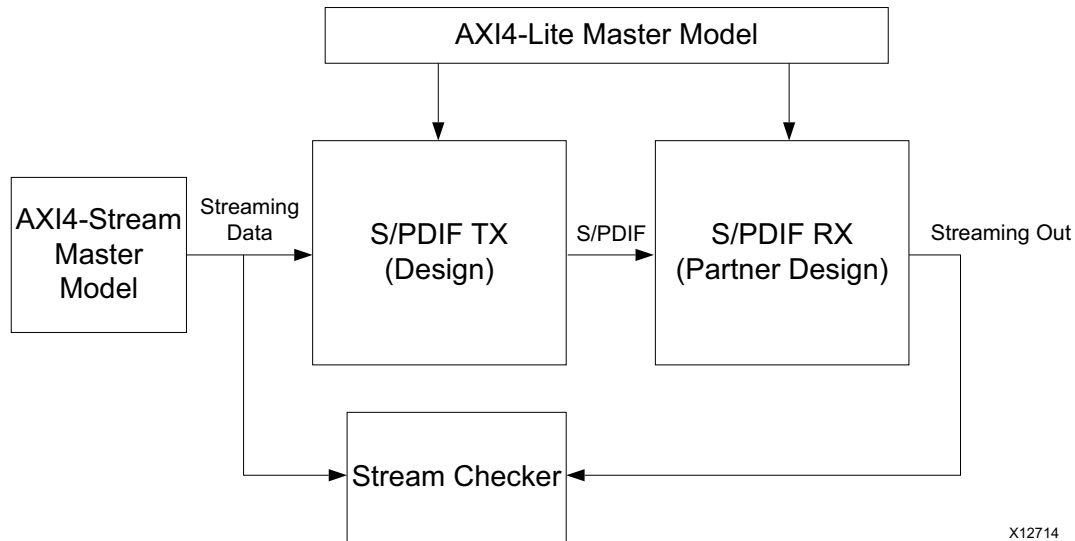


Figure 5-2: SPDIF/AES3 Transmitter Example Design

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

- **SPDIF/AES3 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 SPDIF/AES3 Control register is set to 32'h00000001 to enable the SPDIF/AES3 core in the RX mode. For SPDIF/AES3 TX, the partner Control register is set to 32'h0000000D to enable the SPDIF/AES3 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 SPDIF/AES3 core as defined in the AXI4-Stream protocol. This model generates the 4-bit preambles as required by the SPDIF/AES3 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 SPDIF/AES3 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 SPDIF/AES3 output line) reports whether the data received correctly or any mismatch exists.

---

## SPDIF/AES3 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 SPDIF/AES3 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 SPDIF/AES3 control register is set to `32'h00000001` to enable the SPDIF/AES3 core in RX mode and SPDIF/AES3 TX partner control register is set to `32'h0000000D` to enable the SPDIF/AES3 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 SPDIF/AES3 core as defined in the AXI4-Stream protocol. This model generates the 4-bit preambles as required by the SPDIF/AES3 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 SPDIF/AES3 core with the AXI4-Stream data generated from AXI4-Stream driver.

---

## SPDIF/AES3 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 SPDIF/AES3 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 SPDIF/AES3 control register is set to `32'h0000000D` to enable the SPDIF/AES3 core in TX mode. This configuration sets the TX clock configuration bits to a value of `4'h3` (divisor 24) and the SPDIF/AES3 RX partner control register is set to `32'h00000001` to enable the SPDIF/AES3 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 SPDIF/AES3 core as defined in the AXI4-Stream protocol. This model generates the 4-bit preambles as required by the SPDIF/AES3 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 SPDIF/AES3 RX partner with the AXI4-Stream data generated from AXI4-Stream driver.

# 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/SPDIF:

Xilinx Answer [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 SPDIF/AES3 Transmitter/SPDIF/AES3 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 SPDIF/AES3 Transmitter and `m_axis_aclk` input for SPDIF/AES3 receiver are connected and toggling.
- Check that the AXI4-Stream waveforms are being followed.
- Check core configuration.



## Other Interfaces

In SPDIF/AES3 Receiver mode:

- Check if the SPDIF/AES3 interface line is active or not in case of SPDIF/AES3 Receiver.
- Check the `aud_clk_i` requirement (that is, the clock should be minimum of eight times the SPDIF/AES3 bit rate).
- Check the Status register whether the core capturing the bit rate as expected or not.

In SPDIF/AES3 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 SPDIF/AES3 line rate properly.

# Additional Resources and Legal Notices

---

## Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see [Xilinx Support](#).

---

## Documentation Navigator and Design Hubs

Xilinx® Documentation Navigator provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open the Xilinx Documentation Navigator (DocNav):

- From the Vivado® IDE, select **Help > Documentation and Tutorials**.
- On Windows, select **Start > All Programs > Xilinx Design Tools > DocNav**.
- At the Linux command prompt, enter `docnav`.

Xilinx Design Hubs provide links to documentation organized by design tasks and other topics, which you can use to learn key concepts and address frequently asked questions. To access the Design Hubs:

- In the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the [Design Hubs](#) page.

**Note:** For more information on Documentation Navigator, see the [Documentation Navigator](#) page on the Xilinx website.

---

## 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<sup>®</sup> Design Suite User Guide: Designing with IP* ([UG896](#))
4. *Vivado Design Suite User Guide: Getting Started* ([UG910](#))
5. *Vivado Design Suite User Guide: Logic Simulation* ([UG900](#))
6. *Vivado Design Suite AXI Reference Guide* ([UG1037](#))
7. *ISE<sup>®</sup> 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*
11. AES3 specification ([www.aes.org](http://www.aes.org))

## Revision History

The following table shows the revision history for this document.

| Date       | Version | Revision   |
|------------|---------|--|
| 04/04/2018 | 2.0     | <ul style="list-style-type: none"> <li>Changed the title from S/PDIF to SPDIF/AES3.</li> <li>Added Electrical Circuit Considerations on Board section.</li> <li>Updated Fig. 1-1 SPDIF/AES3 Block Diagram.</li> <li>Added Description to AES3 in the IP Facts</li> </ul>   |
| 11/18/2015 | 2.0     | Added support for UltraScale+ families.  |
| 04/02/2014 | 2.0     | <ul style="list-style-type: none"> <li>Updated Fig. 1-1 S/PDIF Block Diagram.</li> <li>Added Applications section.</li> <li>Added Performance section.</li> <li>Updated description in Resource Utilization section.</li> <li>Updated Example Design and Test Bench chapters.</li> </ul>   |
| 03/20/2013 | 4.0     | <ul style="list-style-type: none"> <li>Updated to core version 2.0 and Vivado Design Suite only.</li> <li>Updated to Questa SIM.</li> <li>Updated Resource Utilization tables.</li> <li>Updated Reset descriptions.</li> <li>Updated Table 3-4 Design Parameters and Table 3-5 Parameter I/O.</li> <li>Added Fig. 4-1 and 4-2 GUI.</li> <li>Updated Output Generation section.</li> <li>Updated code in Clock Frequencies section.</li> <li>Removed Verification appendix.</li> <li>Updated Appendix Debug section.</li> </ul> |
| 12/18/2013 | 2.0     | <ul style="list-style-type: none"> <li>Revision number reverted to 2.0 to align with core version number 2.0.</li> <li>Added IP Integrator and UltraScale support.</li> <li>Added Simulation, Synthesis, Example Design, and Test Bench chapters.</li> <li>Updated Migrating chapter and Debugging Appendix.</li> </ul>  |
| 12/18/2012 | 3.0     | <ul style="list-style-type: none"> <li>Updated to ISE Design Suite 14.4 and Vivado Design Suite 2012.4.</li> <li>Updated description to S/PDIF Control register Bits[5:2].</li> <li>Updated Appendix Debug section.</li> </ul>   |
| 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 <https://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 <https://www.xilinx.com/legal.htm#tos>.

### **AUTOMOTIVE APPLICATIONS DISCLAIMER**

AUTOMOTIVE PRODUCTS (IDENTIFIED AS "XA" IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE ("SAFETY APPLICATION") UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD ("SAFETY DESIGN"). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

© Copyright 2012–2018 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, PrimeCell, and MPCore are trademarks of ARM in the EU and other countries. All other trademarks are the property of their respective owners.

Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения «JONHON», а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов «FORSTAR».



## JONHON

«JONHON» (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

«FORSTAR» (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели,  
кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).



Телефон: 8 (812) 309-75-97 (многоканальный)

Факс: 8 (812) 320-03-32

Электронная почта: [ocean@oceanchips.ru](mailto:ocean@oceanchips.ru)

Web: <http://oceanchips.ru/>

Адрес: 198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А