LogiCORE IP DisplayPort v3.1

DS802 April 24, 2012 **Product Specification**

Introduction

The Xilinx LogiCORE™ IP DisplayPort™ interconnect protocol is designed for transmission and reception of serial-digital video for consumer and professional displays. DisplayPort is a high-speed serial interface standard supported by PC chipsets, GPU's and display controllers, HDTV and monitors from industry leaders and major silicon manufacturers.

This protocol replaces VGA, DVI and HDMI™ outside and LVDS inside the box for higher resolution, higher frame rate and color bit depth display.

DisplayPort IP can support standard rates of 1.62 Gb/s, 2.7 Gb/s and 5.4 Gb/s for consumer and professional displays. When used in high-performance 7 series devices, the DisplayPort core can transmit and receive at 5.4 Gb/s.

The Xilinx DisplayPort IP core is designed to the Video Electronics Standards Association (VESA) *DisplayPort Standard v1.1a and DisplayPort Standard v1.2* specifications [Ref 1][Ref 2].

Features

- Source (TX) and Sink (RX) Controllers
- Designed to VESA DisplayPort Standard v1.1a and v1.2
  - For a 5.4 Gb/s link rate, a high performance FPGA is required with speed grade -2 or -3
  - DisplayPort v1.2 is supported with 7 series devices
- One, two or four pixel-wide video interface supporting up to a 4k x 2k monitor resolution
- 1, 2 or 4 lanes at 1.62, 2.7 or 5.4 Gb/s
- RGB and YCbCr color space, up to 16 bits per color
- Auto lane rate and width negotiation
- I2C over a 1 Mb/s AUX channel
- Secondary channel audio support (2-channel)
- With additional license, supports DisplayPort Audio Support (two-channel with SPDIF). See product page for details.

LogiCORE IP Facts				
Core Specifics				
Supported Device Family(1)	Virtex-7, Kintex-7, Virtex-6, Spartan-6			
Supported User Interfaces	Native Video, AXI4-Stream, AXI4-Lite			
Resources Used				
	I/O (to pins)	LUTs	FFs	Block RAMs
Sink	12	~7000	~5400	0
Source	13	~6500	~5100	0
Provided with Core				
Documentation	Product Specification User Guide			
Design File Formats	Verilog and VHDL NGC Netlist Scripts for Unix and Windows			
Constraints File	.ucf (user constraints file) Full Timing Constraints Transceiver Physical Constraints			
Verification	Verilog Test Bench			
Instantiation Template	Verilog and VHDL Wrapper			
Example Design	Simple RTL Source Policy Maker RTL Sink Policy Maker RTL EDID ROM, RTL I2C Controller Demonstration Test Bench			
Design Tool Requirements				
Xilinx Implementation Tools	ISE 14.1, Vivado 2012.1			
Verification(2)	Mentor Graphics ModelSim Cadence Incisive Enterprise Simulator			
Simulation(2)	Xilinx Synthesis Technology (XST)			
Synthesis	Xilinx XST			
Support				
Provided by Xilinx, Inc.				

1. For a complete listing of supported devices, see the release notes for this core.
2. For the supported versions of the tools, see the ISE Design Suite 14: Release Notes Guide.

Functional Overview

Source Core

The Source core moves a video stream from a standardized main link through a complete DisplayPort Link Layer, and onto High-Speed Serial I/O for transport to a Sink device.

Main Link

The Main Link for the Source core interfaces to a user-driven stream of video data. Using horizontal and vertical sync signals for framing, this user interface matches the industry standard for display controllers and plugs into existing video streams. The user can specify one or two pixel-wide data through a register field. The user can also specify the number of bits per pixel as well as colorspace (RGB or YCbCr or YOnly). In addition, it's possible to specify one, two or four pixel-wide data through a register configuration and provide an accompanying video clock that operates between 13.5 and 150 MHz.

The Source core is responsible for managing the video data and preparing it for transmission over the high-speed serial I/O. It performs the required operations for the Link and Physical Layers of the *DisplayPort Standard v1.1a* or *v1.2*, based on protocol selection. A pre-synthesis directive includes this module, which then can be enabled and disabled through register access. HDCP is not included in the standard CORE Generator output. Contact Xilinx for more details about this feature.

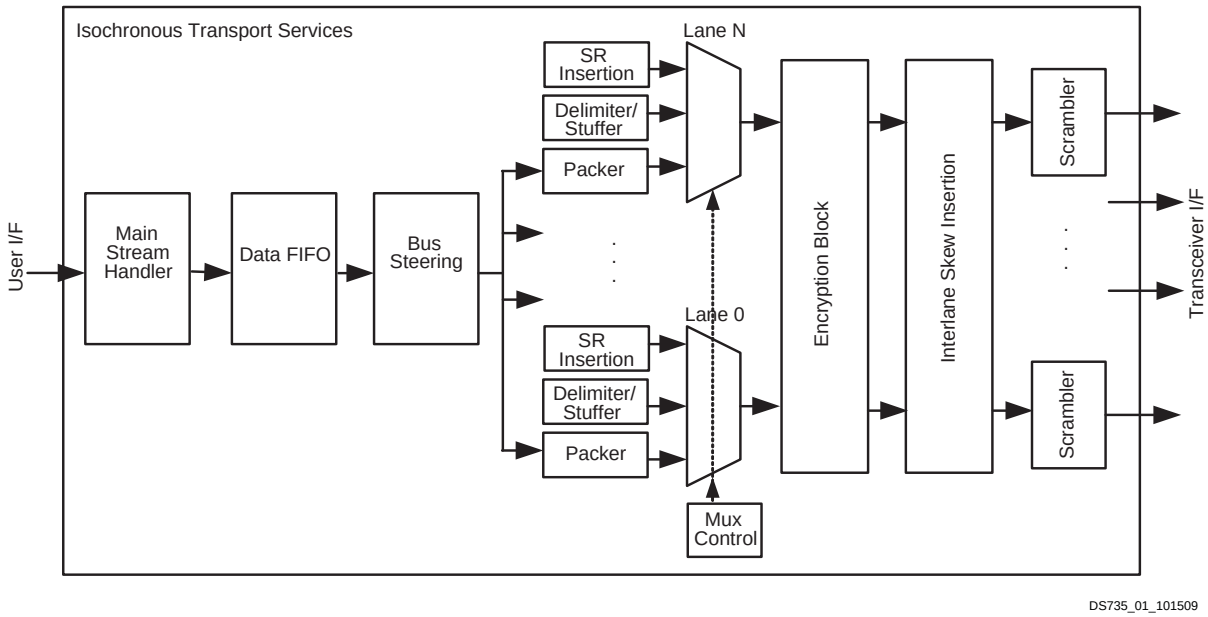

Figure 1: **Source Main Link Datapath**

Secondary Channel

The current version of the DisplayPort IP supports 2-channel Audio. An SPDIF controller is generated when the Audio option is enabled (additional license required). Secondary Channel features from the Displayport v1.1a specification are supported.

The DisplayPort Audio IP core is offered in a modules to provide flexibility and freedom to modify the system as needed. As shown in Figure 2, the Audio interface to the DisplayPort core is defined using an AXI4-Stream interface to improve system design and IP integration.

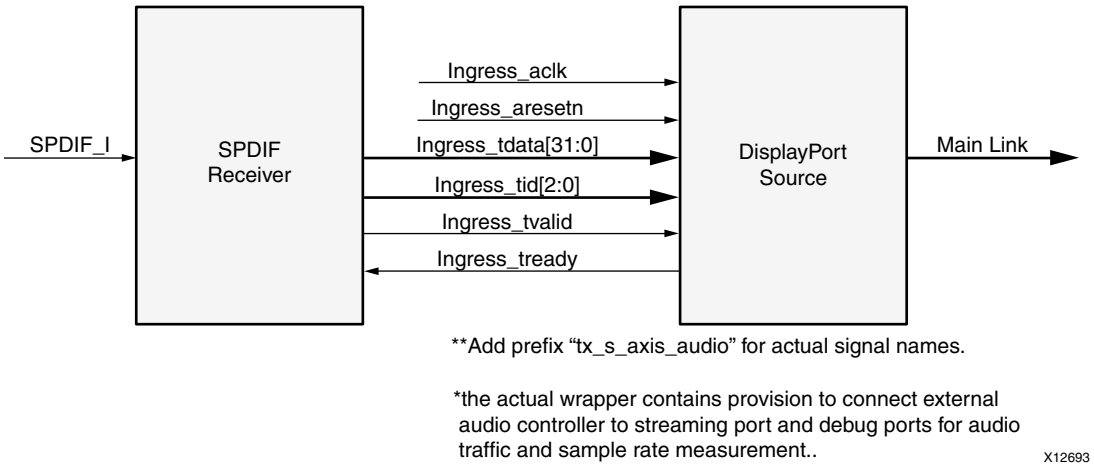

Figure 2: **Audio Data Interface of DisplayPort Source System**

SPDIF is used as the default controller for the DisplayPort Source, and AXI-SPDIF is shipped with the DisplayPort core and delivered in the example design. This system allows access to the AXI4-Stream interface. See the *AMBA AXI4-Stream Specification* for interface timing.

The SPDIF controller as a receiver receives audio samples from the SPDIF line and stores them in an internal buffer. 32-bit AXI TDATA is formatted according as follows:

Control Bits + 24-bit Audio Sample + Preamble

See PG045, *LogiCORE IP SPDIF Product Guide* for more details.

The ingress channel buffer in the DisplayPort core will accept data from the SPDIF controller based on buffer availability and audio control programming. A valid transfer takes place when `tready` and `tvalid` are asserted as described in the AXI4-Stream protocol. The ingress channel buffer acts as a holding buffer.

The DisplayPort Source has a fixed secondary packet length [Header = 4 Bytes + 4 Parity Bytes, Payload = 32 Sample Bytes + 8 Parity Bytes]. In a 1-2 channel transmission, the Source accumulates eight audio samples in the internal channel buffer, and then sends the packet to main link. In a 3-8 channel transmission, the Source waits for at least one sample in all internal channel buffers, and then sends the packet to main link.

Host Interface

The core can be configured through the AMBA® AXI4-Lite processor interface. The registers are mapped as packed 32-bit values from the perspective of the interface.

AUX Channel

The AUX Channel provides peer information between source and sink endpoints. The core is also designed to facilitate I2C communication over this link.

High-Speed Serial I/O

The user can specify up to four lanes through the Xilinx CORE Generator™ GUI. Though more lanes can be selected, the actual number in use is determined by a negotiation procedure between endpoints. The instantiations of the transceivers have been brought to the top and provided to the user for greater visibility.

Sink Core

The Sink device accepts incoming serial data streams from the High-Speed Serial I/O, properly reconstructs the original data to an appropriate format, and provides a video stream in a standard format on the main link. It performs required operations for the Link and Physical Layers of the *DisplayPort Standard v1.1a* and *v1.2* based on protocol selection.

Main Link

The Main Link for the Sink Core drives a stream of video data toward the user. Using horizontal and vertical sync signals for framing, this user interface matches the industry standard for display controllers and plugs in to existing video streams with little effort. Though the core provides data and control signaling, the user is still expected to supply an appropriate clock. This clock can be generated with the use of M and N values provided by the core. Alternatively, the user might want to generate a clock by other means. The core's underflow protection allows the user to use a fast clock to transfer data into a frame buffer.

The user can specify one, two, or four pixel-wide data through a register field. The bit width and format is determined from the Main Stream Attributes, which are provided as register fields.

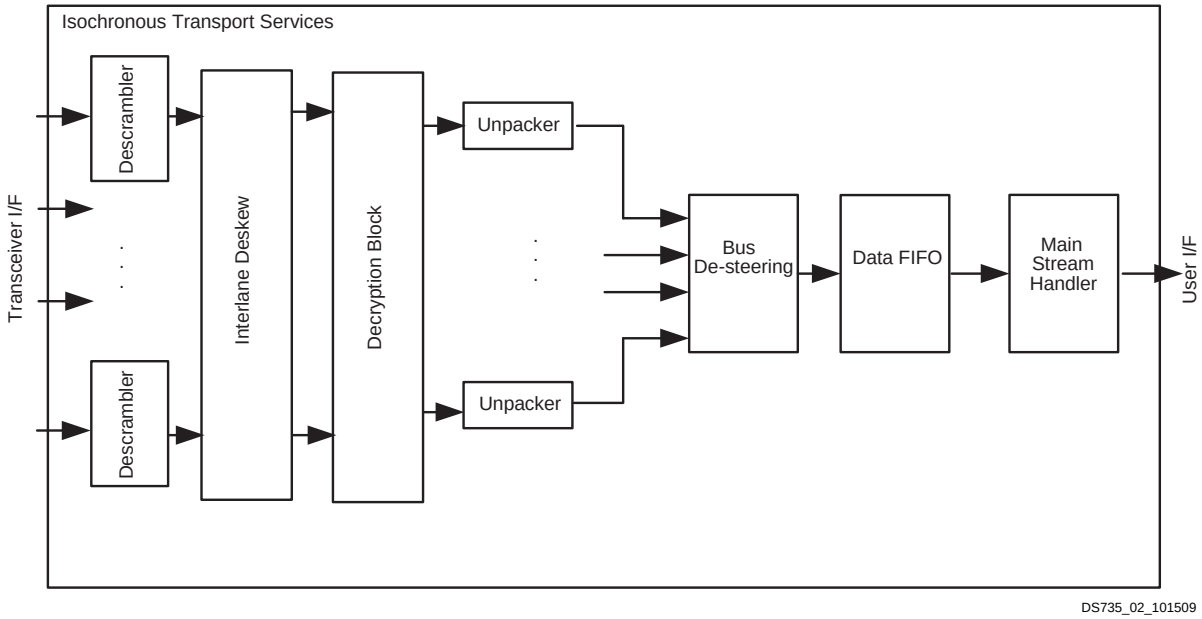

Figure 3: **Sink Main Link Datapath**

Secondary Channel

The current version of the DisplayPort core supports two-channel Audio. The SPDIF controller is generated when the Audio option is enabled. Secondary Channel features from the Displayport v1.1a specification are supported. DisplayPort Audio IP core is offered in modules to provide flexibility to modify the system as needed.

As shown in Figure 4, Audio interface to DisplayPort is defined using AXI4-Stream interface.

SPDIF is used as default controller for DisplayPort sink and SPDIF is shipped along with DisplayPort IP and delivered in example design. User will have access to AXI-Streaming interface. See the *AMBA AXI-Streaming Specification* for interface timing.

Audio data and secondary packets are received from the main link and stored in internal buffers of DisplayPort Sink core. The AXI4-Stream interface of DisplayPort transfers audio sample along with control bits to SPDIF

transmitter and AXI4-Stream slave has to accept it immediately. In other words, DisplayPort Sink should never be back pressured.

SPDIF transmitter sends out samples as per SPDIF protocol format. Typically SPDIF PHY is a differential device and hence user should create a differential signal and make proper connections to PHY at system level.

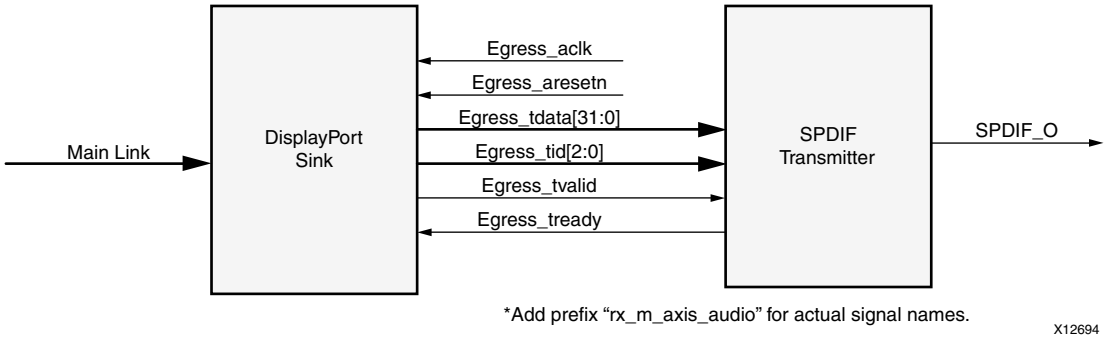

Figure 4: **Audio Data Interface of DisplayPort Sink System**

Host Interface

The core can be configured through the AMBA AXI4-Lite processor interface. The registers are mapped as packed 32-bit values from the perspective of the interface.

AUX Channel

The AUX Channel provides peer information between source and sink endpoints. The core is also designed to facilitate I2C communication over this link.

High-Speed Serial I/O

The user can specify up to four lanes through a pre-synthesis directive. Though more lanes can be selected, the actual number in use is determined by a negotiation procedure between endpoints. The instantiations of the transceivers have been brought to the top and provided to the user for greater visibility.

Policy Maker System Example Reference Design

Xilinx provides both Source and Sink DisplayPort controllers. The RTL-based Sink Policy Maker is suitable for core bring up and link maintenance. It enables the core, asserts Hot Plug Detect, and enables the Display Timing Generator. For users who desire finer tuning of the link, this Policy Maker is provided as open source.

Two Source Policy Maker options are available to the purchaser of the DisplayPort core. The first is a simple RTL-based controller. Use this version for a quick simulation or fast training against the Xilinx Sink DisplayPort core in hardware. As with the Sink version, this is provided as open source for user tuning. This version should not be used in applications that require compliance to the VESA specification.

For full compliance, Xilinx provides a second netlist-based Source Policy Maker. This version is ideal for hardware bring up and productization. It is based off of a MicroBlaze™ design running C code, which is also available to the purchaser of this core. For information about acquiring the source code, contact your local Xilinx sales office. See Ordering Information, page 13 for more details.

XAPP493, *Implementing a DisplayPort Source Policy Maker Using a MicroBlaze Embedded Processor*, describes the implementation of a DisplayPort Source Policy Maker targeted specifically for the Spartan-6 FPGA Consumer Video Kit (CVK). XAPP593, *Displayport Sink Reference Design*, describes the implementation of a DisplayPort

Source/Sink Policy Maker targeted specifically for the Spartan-6 FPGA Consumer Video Kit (CVK1.0). Both documents can be found on xilinx.com.

I/O Signals

Table 1: **Source Core I/O Signals**

Signal Name(1)	Direction From Core	Description
DisplayPort Processor Interface		
s_axi_aclk	Input	AXI Bus Clock.
s_axi_aresetn	Input	AXI Reset. Active low.
s_axi_awaddr[31:0]	Input	Write Address.
s_axi_awprot[2:0]	Input	Protection type.
s_axi_awvalid	Input	Write address valid.
s_axi_awready	Output	Write address ready.
s_axi_wdata[31:0]	Input	Write data bus.
s_axi_wstrb[3:0]	Input	Write strobes.
s_axi_wvalid	Input	Write valid.
s_axi_wready	Output	Write ready.
s_axi_bresp[1:0]	Output	Write response.
s_axi_bvalid	Output	Write response valid.
s_axi_bready	Input	Response ready.
s_axi_araddr[31:0]	Input	Read address.
s_axi_arprot[2:0]	Input	Protection type.
s_axi_arvalid	Input	Read address valid.
s_axi_arready	Output	Read address ready.
s_axi_rdata[31:0]	Output	Read data.
s_axi_rresp[1:0]	Output	Read response.
s_axi_rvalid	Output	Read valid.
s_axi_rready	Input	Read ready.
axi_int	Output	AXI interrupt out.
User Data Interface		
tx_vid_clk	Input	User data video clock.
tx_vid_vsync	Input	Vertical sync pulse.
tx_vid_hsync	Input	Horizontal sync pulse.
tx_vid_oddeven	Input	Odd/even field select.
tx_vid_enable	Input	User data video enable.
tx_vid_pixel0[47:0]	Input	Video data.
tx_vid_pixel1[47:0]	Input	Video data.

Table 1: **Source Core I/O Signals *(Cont'd)***

Signal Name[1]	Direction From Core	Description
tx_vid_pixel2[47:0]	Input	Video data.
tx_vid_pixel3[47:0]	Input	Video data.
tx_vid_rst	Input	User video reset.
Main Link Interface		
lnk_clk_p	Input	Differential clock input from pin.
lnk_clk_n	Input	Differential clock input from pin.
lnk_clk	Output	Reference clock for the FPGA fabric.
lnk_tx_lane_p[3:0]	Output	High-speed lane serial data.
lnk_tx_lane_n[3:0]	Output	High-speed lane serial data.
AUX Channel Interface		
aux_tx_channel_in_p	Input	Positive polarity of the AUX Manchester-II data.
aux_tx_channel_in_n	Input	Negative polarity of the AUX Manchester-II data.
aux_tx_channel_out_p	Output	Positive polarity of the AUX Manchester-II data.
aux_tx_channel_out_n	Output	Negative polarity of the AUX Manchester-II data.
aux_tx_io_p	Input/Output	Positive Polarity AUX Manchester-II data (used for Spartan-6 devices)
aux_tx_io_n	Input/Output	Negative Polarity AUX Manchester-II data (used for Spartan-6 devices)
HPD Interface		
tx_hpd	Input	Hot Plug Detect.
SPDIF Audio Processor Interface		
aud_s_axi_aclk	input	AXI Bus Clock
aud_s_axi_aresetn	input	AXI Reset. Active low.
aud_s_axi_awaddr[31:0]	input	Write Address.
aud_s_axi_awprot[2:0],	input	Protection type.
aud_s_axi_awvalid	Input	Write address valid.
aud_s_axi_awready	Output	Write address ready.
aud_s_axi_wdata[31:0]	Input	Write data bus.
aud_s_axi_wstrb[3:0]	Input	Write strobes.
aud_s_axi_wvalid	Input	Write valid.
aud_s_axi_wready	Output	Write ready.
aud_s_axi_bresp[1:0]	Output	Write response.
aud_s_axi_bvalid	Output	Write response valid
aud_s_axi_bready	Input	Response ready.
aud_s_axi_araddr[31:0]	Input	Read address.
aud_s_axi_arprot[2:0]	Input	Protection type.
aud_s_axi_arvalid	Input	Read address valid.

Table 1: **Source Core I/O Signals *(Cont'd)***

Signal Name[1]	Direction From Core	Description
aud_s_axi_arready	Output	Read address ready.
aud_s_axi_rdata[31:0]	Output	Read data.
aud_s_axi_rresp[1:0]	Output	Read response.
aud_s_axi_rvalid	Output	Read valid.
aud_s_axi_rready	Input	Read ready.
aud_axi_int	Output	AXI interrupt out.
SPDIF Interface		
spdif_in	Input	SPDIF channel input.
Audio Clock Interface		
aud_clk	Input	Audio sample clock (512 * fs). fs= sampling frequency.
aud_rst	Input	Audio Interface Reset (Active High).
aud_axis_aclk	Input	Audio streaming interface clock (greater than or equal to 512 * fs)
aud_axis_aresetn	Input	Audio Streaming Interface Reset (Active Low).
spdif_sample_clk	Input	SPDIF Controller sampling clock. Should be greater than or equal to 512*fs.

1. Signal names beginning with s_ or m_ denote slave and master interfaces respectively.

Table 2: **Sink Core I/O Signals**

Signal Name[1]	Direction From Core	Description
DisplayPort Processor Interface		
s_axi_aclk	Input	AXI Bus Clock .
s_axi_aresetn	Input	AXI Reset. Active low.
s_axi_awaddr[31:0]	Input	Write Address.
s_axi_awprot[2:0]	Input	Protection type.
s_axi_awvalid	Input	Write address valid.
s_axi_awready	Output	Write address ready.
s_axi_wdata[31:0]	Input	Write data bus.
s_axi_wstrb[3:0]	Input	Write strobes.
s_axi_wvalid	Input	Write valid.
s_axi_wready	Output	Write ready.
s_axi_bresp[1:0]	Output	Write response.
s_axi_bvalid	Output	Write response valid.
s_axi_bready	Input	Response ready.
s_axi_araddr[31:0]	Input	Read address.
s_axi_arprot[2:0]	Input	Protection type.

Table 2: **Sink Core I/O Signals *(Cont'd)***

Signal Name[1]	Direction From Core	Description
s_axi_arvalid	Input	Read address valid.
s_axi_arready	Output	Read address ready.
s_axi_rdata[31:0]	Output	Read data.
s_axi_rresp[1:0]	Output	Read reponse.
s_axi_rvalid	Output	Read valid.
s_axi_rready	Input	Read ready.
axi_int	Output	AXI interrupt out.
User Data Interface		
rx_vid_clk	Input	User data video clock.
rx_vid_vsync	Output	Vertical sync pulse.
rx_vid_hsync	Output	Horizontal sync pulse.
rx_vid_oddeven	Output	Odd/even field select.
rx_vid_enable	Output	User data video enable.
rx_vid_pixel0[47:0]	Output	Video data.
rx_vid_pixel1[47:0]	Output	Video data.
rx_vid_pixel2[47:0]	Output	Video data.
rx_vid_pixel3[47:0]	Output	Video data.
rx_vid_rst	Input	User video reset.
Main Link Interface		
lnk_clk	Output	Reference clock for the FPGA fabric.
lnk_clk_p	Input	Differential clock input from pin.
lnk_clk_n	Input	Differential clock input from pin.
lnk_rx_lane_p[3:0]	Input	High-speed lane serial data.
lnk_rx_lane_n[3:0]	Input	High-speed lane serial data.
lnk_m_vid[23:0]	Output	M-value for clock generation.
lnk_n_vid[23:0]	Output	N-value for clock generation.
lnk_m_aud[23:0]	Output	M-value for audio clock generation.
lnk_n_aud[23:0]	Output	N-Value for audio clock generation.
AUX Channel Interface		
aux_rx_channel_in_p	Input	Positive polarity of the AUX Manchester-II data.
aux_rx_channel_in_n	Input	Negative polarity of the AUX Manchester-II data.
aux_rx_channel_out_p	Output	Positive polarity of the AUX Manchester-II data.
aux_rx_channel_out_n	Output	Negative polarity of the AUX Manchester-II data.
aux_rx_io_p	Input/Output	Positive Polarity AUX Manchester-II data (used for Spartan-6 devices).
aux_rx_io_n	Input/Output	Negative Polarity AUX Manchester-II data (used for Spartan-6 devices).

Table 2: **Sink Core I/O Signals *(Cont'd)***

Signal Name[1]	Direction From Core	Description
I2C Interface		
i2c_sda_in	Input	I2C serial data in.
i2c_sda_enable_n	Output	I2C data out enable. Active low.
i2c_scl_in	Input	I2C serial clock in.
i2c_scl_enable_n	Output	I2C serial clock output enable. Active low.
HPD Interface		
rx_hpd	Output	Hot Plug Detect.
SPDIF Audio Processor Interface		
aud_s_axi_aclk	Input	AXI Bus Clock.
aud_s_axi_aresetn	Input	AXI Reset. Active Low.
aud_s_axi_awaddr[31:0]	Input	Write Address.
aud_s_axi_awprot[2:0]	Input	Protection type.
aud_s_axi_awvalid	Input	Write address valid.
aud_s_axi_awready	Output	Write address ready.
aud_s_axi_wdata[31:0]	Input	Write data bus.
aud_s_axi_wstrb[3:0]	Input	Write strobes.
aud_s_axi_wvalid	Input	Write valid.
aud_s_axi_wready	Output	Write ready.
aud_s_axi_bresp[1:0]	Output	Write response.
aud_s_axi_bvalid	Output	Write response valid
aud_s_axi_bready	Input	Response ready.
aud_s_axi_araddr[31:0]	Input	Read address.
aud_s_axi_arprot[2:0]	Input	Protection type.
aud_s_axi_arvalid	Input	Read address valid.
aud_s_axi_arready	Output	Read address ready.
aud_s_axi_rdata[31:0]	Output	Read data.
aud_s_axi_rresp[1:0]	Output	Read response.
aud_s_axi_rvalid	Output	Read valid.
aud_s_axi_rready	Input	Read ready.
aud_axi_int	Output	AXI interrupt out. AXI interrupt out of SPDIF controller.
Audio Clock Interface		
aud_clk	Input	Audio sample clock (512 * fs). fs= sampling frequency.
aud_rst	Input	Audio Interface Reset (Active High).
aud_axis_aclk	Input	Audio streaming interface clock (greater than or equal to 512 * fs)

Table 2: **Sink Core I/O Signals *(Cont'd)***

Signal Name(1)	Direction From Core	Description
aud_axis_aresetn	Input	Audio Streaming Interface Reset (Active Low).
SPDIF Interface		
spdif_out	Output	SPDIF channel output

1. Signal names beginning with s_ or m_ denote slave and master interfaces respectively.

Parameterization

Table 3: **Core Parameters**

Parameter Name	Values (Default)	Description
PROTOCOL_SELECTION	0	Protocol selection: • 0: DisplayPort v1.1a • 1: DisplayPort v1.2
LINK_RATE	{1.62,2.7,5.4}	Maximum Link Rate supported
LANE COUNT	{1, 2, **4**}	Number of high-speed SerDes the design contains.
SECONDARY_SUPPORT	0	Enables secondary channel logic to send Audio packets.
AUDIO_CHANNELS	2	Current version of IP supports 2-channel audio. The value is hard coded.
MAX_BITS_PER_COLOR	8	Sets maximum bits per color support and optimizes IP accordingly.
QUAD_PIXEL_ENABLE	0	Enables support of quad-pixel video interface.
DUAL_PIXEL_ENABLE	1	Enables support of dual-pixel video interface.
YCRCB_ENABLE	1	Enables YCrCb 4:2:2 colorimetry support.
YONLY_ENABLE	0	Enables Y-Only colorimetry support
VENDOR_SPECIFIC	0	Enables DPCD space of Vendor-Specific fields in Sink.
IEEE OUI	24-bit value	This Sink-only parameter allows the user to hardwire the OUI value within the core.

Verification

The DisplayPort cores have been verified with functional simulation and hardware testing.

Simulation

A highly-parameterizable transaction-based test bench was used to test the core. Broad protocol and implementation-specific coverage were used to fully verify the cores. The tests included the following:

- Full I2C operation over the AUX channel
- Bandwidth and performance tests
- Main link stress tests
- Processor interface register read and write accesses
- Scramble/Descramble quality checks

- Video and Audio data integrity checks

Hardware Validation

The DisplayPort cores have been validated using DNMEG_V5T_PCIE boards from The Dini Group and a Spartan-6 FPGA Consumer Video Kit (CVK) from TED. The hardware has been tested against Quantum Data's 882 Test Instrument for compliance to the DisplayPort standard. The hardware has been tested against Quantum Data's 882 Test Instrument for Link layer compliance to the DisplayPort v1.1a standard. The cores were also tested against established, certified products for interoperability.

Family Support

The cores operate at DisplayPort v1.1a speed (1.62 Gb/s and 2.7 Gb/s) with Virtex-6 -1, Spartan-6 -3 or XA Spartan-6 -3 parts. The cores operate at DisplayPort v1.2 speed (5.4 Gb/s) with 7 series -2 or -3 parts. The supported device families are:

- Virtex-7
- Kintex-7
- Virtex-6 LXT
- Virtex-6 SXT
- Virtex-6 HXT
- Spartan-6 LXT
- XA Spartan-6 LXT

References

1. VESA *DisplayPort Standard v1.1a*, January 11, 2008.
2. VESA *DisplayPort Standard v1.2*, December 22, 2009.
3. UG386, *Spartan-6 FPGA GTP Transceivers User Guide*.
4. UG371, *Virtex-6 FPGA GTH Transceivers User Guide*.
5. UG366, *Virtex-6 FPGA GTX Transceivers Advance Product Specification*.
6. Spartan-6 FPGA Consumer Video Kit
7. XAPP493, *Implementing a DisplayPort Source Policy Maker Using a MicroBlaze Embedded Processor.*
8. High-bandwidth Digital Content Protection System v1.3 Amendment for DisplayPort, v1.0
9. AMBA AXI Protocol, v2.0
10. UG476, *7 Series FPGAs GTX Transceivers User Guide*
11. UG761, *Xilinx AXI Reference Guide*

Support

Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled *DO NOT MODIFY*.

Ordering Information

This Xilinx LogiCORE IP module is provided under the terms of the Xilinx Core Site License. The core is generated using the CORE Generator software provided with the Xilinx ISE® Design Suite.

To evaluate this core at no charge, refer to the evaluation instructions on the DisplayPort product page, located at www.xilinx.com/products/ipcenter/EF-DI-DISPLAYPORT.htm. To access the full core functionality in simulation and in hardware, you must purchase and install a full license for the core.

Contact your local Xilinx sales representative for information on pricing and availability of this and other Xilinx LogiCORE IP modules. Information about additional modules can be found at the Xilinx IP Center.

Revision History

The following table shows the revision history for this document.

Date	Version	Description of Revisions
9/21/10	1.0	Initial Xilinx release.
3/1/2011	2.0	Updated core to v2.2 and ISE Design Suite to v13.1.
6/22/2011	3.0	Updated core to v2.3 and ISE Design Suite to v13.2. Added support for Virtex-7 and Kintex-7 devices. Removed support for Virtex-5 devices. Amended HDCP support.
4/24/2012	4.0	Updated core to v3.1 and ISE Design Suite to v14.1. Added support for DisplayPort Standard v1.2 for 7 series devices.

Notice of Disclaimer