DisplayPort TX Subsystem v1.0

Product Guide

Vivado Design Suite

PG199 November 18, 2015

XILINX
ALL PROGRAMMABLE™

Table of Contents

Introduction

DisplayPort TX Subsystem is a plug-in solution for serial digital data transmission in large video systems of up to video resolutions of 4k2k at 60fps. The subsystem provides ease of use in selecting the required mode and the rest of the customization is automated.

Features

- Support for DisplayPort Source (TX) transmissions.
- Supports multi-stream transport (MST) and single stream transport (SST) at Ultra HD at 60 fps
- Dynamic link rate support (1.62/2.7/5.4 Gb/s)
- Dynamic support of 6, 8, 10, 12, or 16 bits per color (BPC).
- Dynamic support of RGB/YCbCr444/ YCbCr422/Y_Only color formats.
- Wide screen support with internal split of up to two streams of the same resolution.
- Support 32-bit Video PHY(GT) Interface
- Supports 2 to 8 channel Audio.
- Supports HDCP(1.3) encryption.

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	UltraScale+™ Families, UltraScale™ Architecture, Zynq®-7000, 7 Series
Supported User Interfaces	AXI4-Stream, AXI4-Lite
Resources	Performance and Resource Utilization web page
Provided with Core	
Design Files	Hierarchical subsystem packaged with DisplayPort TX core and other IP cores
Example Design	N/A
Test Bench	N/A
Constraints File	IP cores delivered with XDC files
Simulation Model	N/A
Supported S/W Driver	Standalone
Tested Design Flows(2)	
Design Entry	Vivado® Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide.
Synthesis	Vivado Synthesis
Support	
Provided by Xilinx at the Xilinx Support web page	

Notes:

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

Overview

This chapter contains an overview of the core as well as details about features, licensing, and standards. The DisplayPort TX Subsystem is a full feature, hierarchically packaged subsystem with a DisplayPort Transmit (TX) core ready to use in applications in large video systems.

Feature Summary

- UHD up to 60 fps supports multi-stream transport (MST) and SST modes.
- Dynamic support of different bits per color (6, 8, 10, 12 or 16) and line rates.
- Dynamic support of RGB/YCbCr444/ YCbCr422/Y_Only color formats.
- Support optional HDCP Controller.

Unsupported Features

- MST Audio is not supported.
- HDCP is not supported in MST mode.
- Output Video Streaming interface is not scalable with dynamic pixel mode selection.
- 16-Bit Video PHY interface is not supported.

Licensing and Ordering Information

This section contains information about licensing the core.

License Checkers

If the IP requires a license key, the key must be verified. The Vivado® design tools have several license checkpoints for gating licensed IP through the flow. If the license check succeeds, the IP can continue generation. Otherwise, generation halts with error. License checkpoints are enforced by the following tools:

- Vivado Synthesis
- Vivado Implementation
- write_bitstream (Tcl command)

IMPORTANT: *IP license level is ignored at checkpoints. The test confirms a valid license exists. It does not check IP license level.*

License Type

This subsystem requires a license for the DisplayPort Transmit core, which is provided under the terms of the Xilinx Core License Agreement. 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 of Xilinx LogiCORE IP.

For more information about licensing for the core, see the DisplayPort product 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.

Product Specification

This chapter contains a high-level overview of the core as well as performance and port details.

Overview

The DisplayPort TX Subsystem operates in the following video modes:

- Single stream transport (SST)
- Multi-stream transport (MST)

In the SST mode, the subsystem is packaged with three subcores: DisplayPort Transmit core, Video Timing Controller (VTC) and DP AXI4-Stream to Video Bridge. In the SST mode, the TX subsystem also includes optional HDCP controller for encryption and AXI Timer as a helper core for HDCP functionality.

Because the DisplayPort TX Subsystem is hierarchically packaged, you select the parameters and the subsystem creates the required hardware. Figure 2-1 shows the architecture of the subsystem assuming MST with four streams.

The subsystem includes a multi-pixel AXI4-Stream Video Protocol interface. The DisplayPort TX Subsystem outputs the video using the DisplayPort v1.2 protocol. The DisplayPort TX Subsystem works in conjunction with Video PHY Controller configured for the DP protocol.

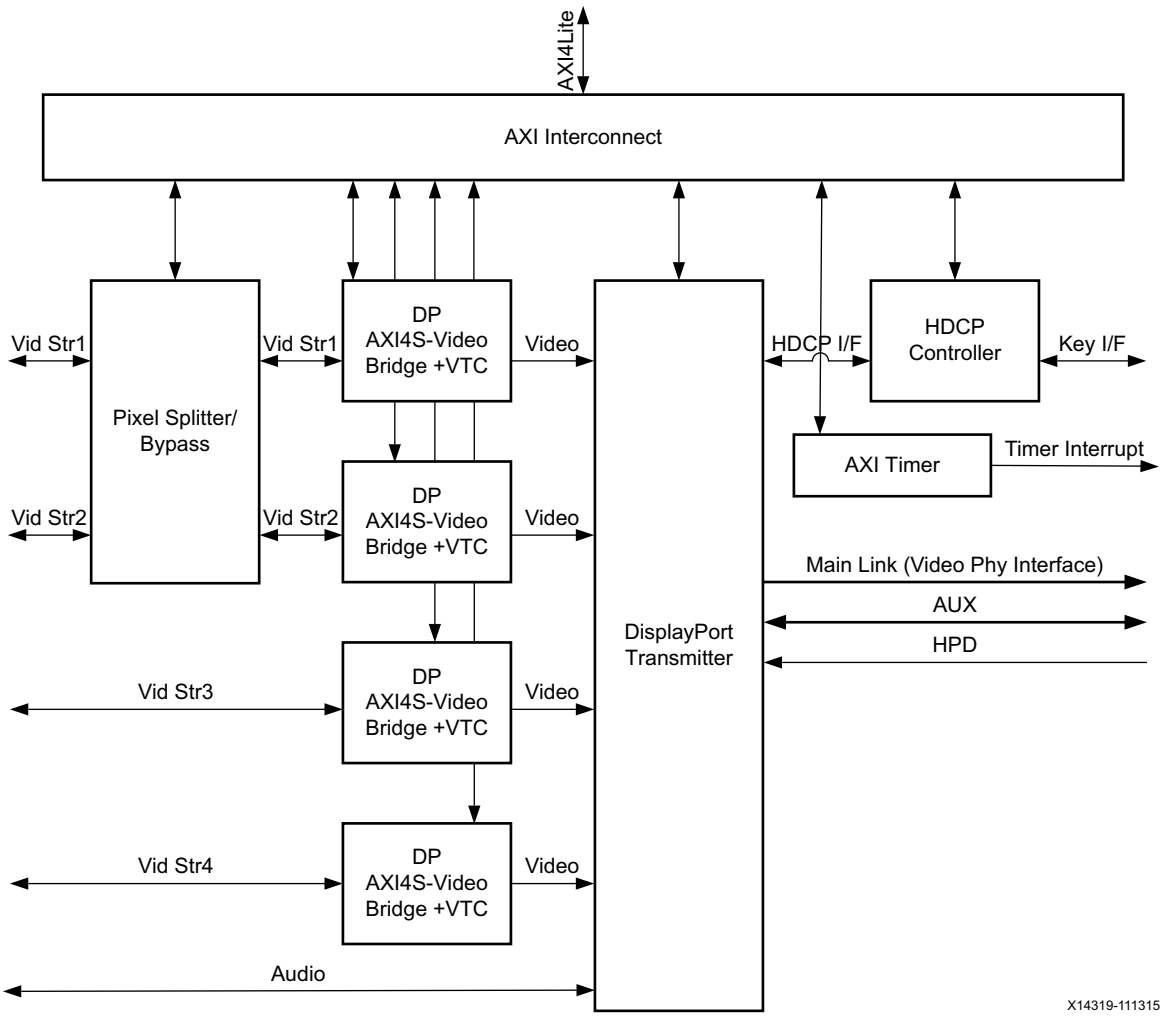

Figure 2-1: **DisplayPort TX Subsystem Block Diagram**

In the MST mode, the subsystem has four subcores: Dual Splitter, DisplayPort AXI4-Stream to Video Bridge, Video Timing Controller and DisplayPort Transmitter core.

DisplayPort Dual Splitter

The Dual Splitter is used to vertically split the frame to support MST with two streams, as shown in Figure 2-2. Despite the frames being split, you will see this as one frame. The Dual Splitter has a buffer to hold the data for up to one and a half scan lines.

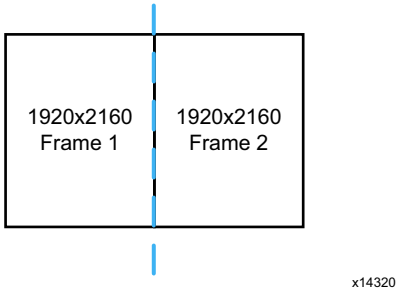

Figure 2-2: **Vertically Split Frame**

Splitter Interface

The splitter input and output are video over AXI4-Stream interface. Figure 2-3 shows the timing of this interface.

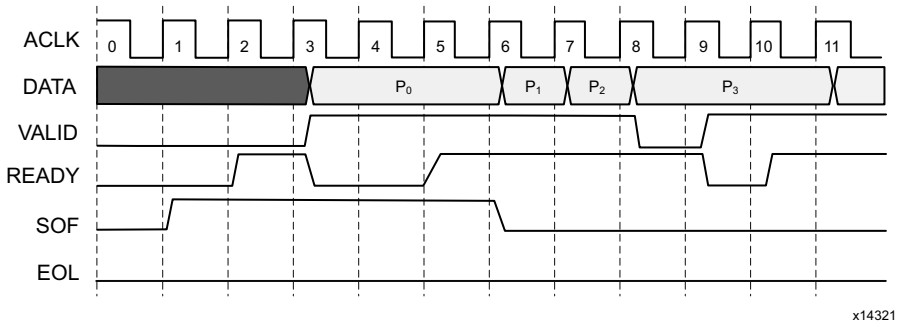

Figure 2-3: **Video over AXI4-Stream Interface Timing**

Based on the mode, the Core Control register (CORE_CONTROL_REG) of the Dual Splitter must be configured for input and output samples per clock. See Dual Splitter Registers for a description of CORE_CONTROL_REG.

DisplayPort AXI4-Stream to Video Bridge

The DisplayPort AXI4-Stream to Video Bridge maps the video over the AXI4-Stream interface to native video format as required by the DisplayPort Transmit IP core. The bridge uses the Xilinx AXI4 to Video Out core to convert the format between AXI4-Stream to DisplayPort native video.

For details about the Video Out Bridge, see the *AXI4-Stream to Video Out Product Guide* (PG044) [Ref 10].

For details about the video over AXI4-Stream, see the *AXI Reference Guide* (UG1037) [Ref 8].

The *DisplayPort Product Guide* (PG064) contains information about the DisplayPort input video format [Ref 9].

The receive side of the bridge is Video over AXI4-Stream. For more details, see Port Descriptions.

In MST mode, there are N number of bridges in the subsystem, where N = the number of AXI4-Stream inputs to the subsystem.

Video Timing Controller

The Xilinx Video Timing Controller is used for generation of video timing. For details on this core, see the *Video Timing Controller Product Guide* (PG016) [Ref 11].

IMPORTANT: *You must program proper front porch and back porch blanking period generation.*

DisplayPort Transmit

The DisplayPort Transmit core contains the following components as shown in Figure 2-4:

- **Main Link**: Provides delivery of the primary video stream.
- **Secondary Link**: Integrates the delivery of audio information into the Main Link blanking period.
- **AUX Channel**: Establishes the dedicated source to sink communication channel.

For more details, see the *DisplayPort Product Guide* (PG064) [Ref 9].

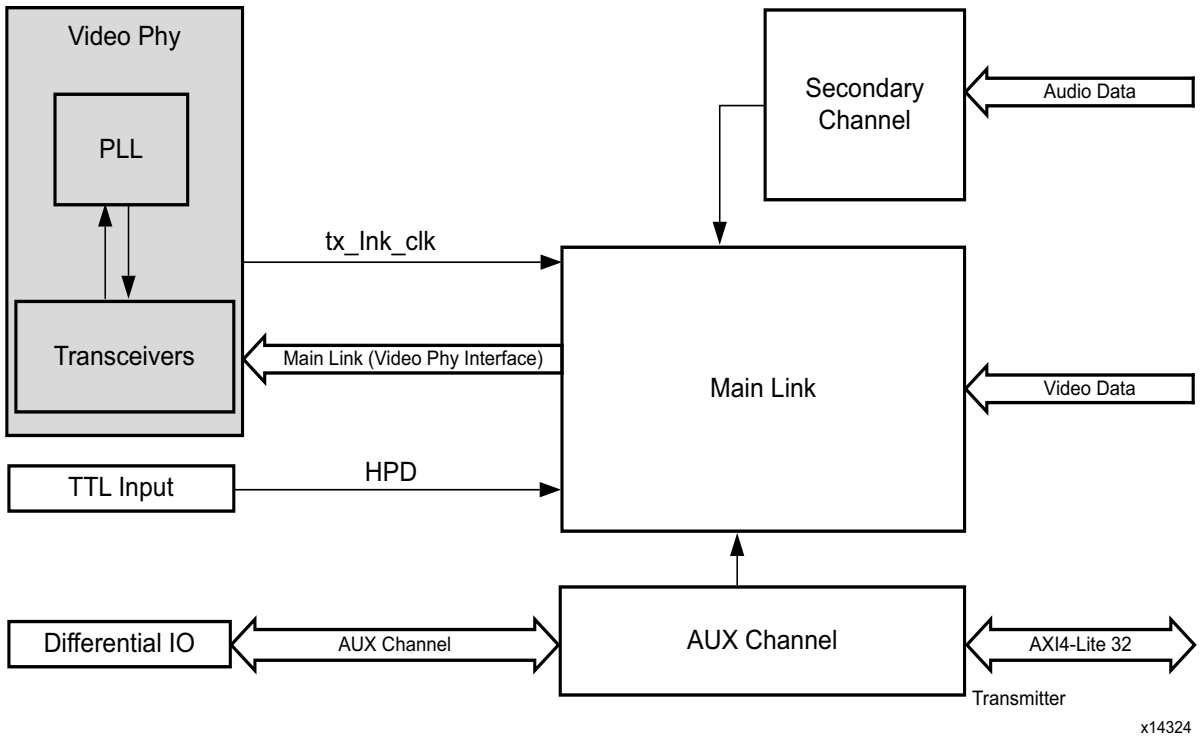

Figure 2-4: **DisplayPort Transmit Core Block Diagram**

AXI Interconnect

The subsystem uses Xilinx AXI Interconnect IP core, as a crossbar which contains an AXI4-Lite interface. Figure 2-5 shows the AXI slave structure within the DisplayPort TX Subsystem.

Note: For MST with *N* streams, there are *N* Video Timing Controllers. See Address Map Example in Chapter 3.

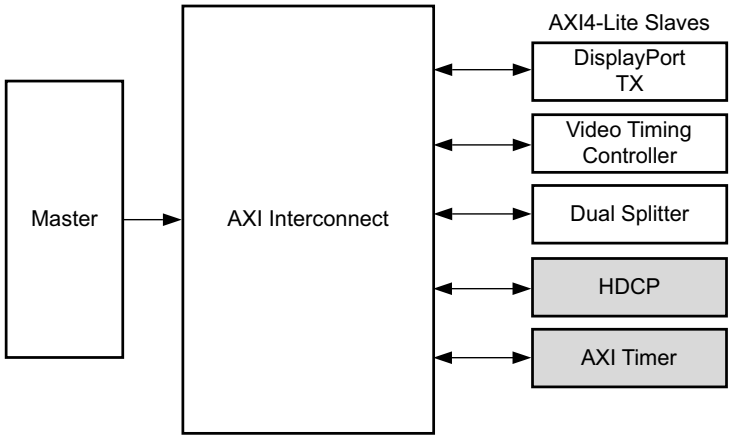

Figure 2-5: **AXI4-Lite Interconnect within DisplayPort TX Subsystem**

HDCP Controller

The HDCP v1.3 protocol specifies a secure method of transmitting audiovisual content. Further, the audiovisual content can be transmitted over a DisplayPort interface. HDCP Controller is used for data encryption along with Displayport transmit IP in Displayport TX subsystem.

Figure 2-6 shows the DisplayPort TX Subsystem with HDCP controller. For more details on HDCP, see the *HDCP Controller Product Guide* (PG224) [Ref 12].

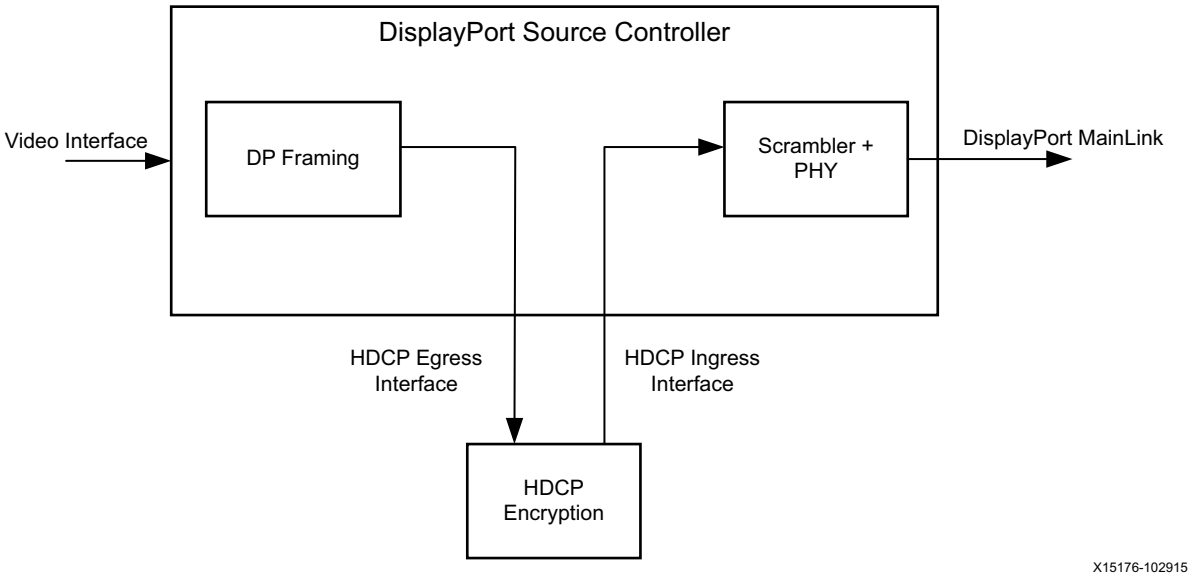

Figure 2-6: **DisplayPort TX with HDCP Controller**

AXI Timer

A 32- Bit AXI Timer is used in Displayport TX subsystem when the HDCP controller is enabled for encryption. The AXI Timer can be accessed through AXI4 master interface for basic timer functionality in the system.

Standards

The DisplayPort TX Subsystem is compatible with the DisplayPort v1.2 Standard, HDCP v1.3 standard, as well as the AXI4-Lite and AXI4-Stream interfaces.

Resource Utilization

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

Port Descriptions

The DisplayPort TX Subsystem ports are described in Table 2-1.

Table 2-1: **DisplayPort TX Subsystem Ports**

Signal Name	Direction from Core	Description
AXI4-Lite Interface		
s_axi_aclk	Input	AXI Bus Clock.
s_axi_aresetn	Input	AXI Reset. Active-Low.
s_axi_awaddr[18:0]	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[18:0]	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.
AXI4-Stream Interface		
s_axis_aclk_stream1	Input	AXI4-Stream clock.
s_axis_aresetn_stream1	Input	AXI4-Stream reset. Active-Low.
s_axis_video_stream1_tdata[191:0]	Input	Video data input.
s_axis_video_stream1_tlast	Input	Video end of line.
s_axis_video_stream1_tready	Output	AXI4-Stream tready output.
s_axis_video_stream1_tuser	Input	Video start of frame.
s_axis_video_stream1_tvalid	Input	Video valid.

Table 2-1: **DisplayPort TX Subsystem Ports *(Cont'd)***

Signal Name	Direction from Core	Description
MST Stream (*n* = stream number 2 to 4) ***Note:*** See Clocking in Chapter 3 for the clock values.		
s_axis_aclk_stream*n*	Input	MST stream clock.
s_axis_aresetn_stream*n*	Input	MST stream reset. Active-Low.
s_axis_video_stream*n*_tdata[191:0]	Input	MST stream video data input.
s_axis_video_stream*n*_tlast	Input	MST stream video end of line.
s_axis_video_stream*n*_tready	Output	MST stream input ready.
s_axis_video_stream*n*_tuser	Input	MST stream video start of frame.
s_axis_video_stream*n*_tvalid	Input	MST stream video valid.
m_aclk_stream1	Input	Video pipe clock for stream1. Used in MST configuration.
m_aresetn_stream1	Input	Active-Low video pipe reset for stream 1. Used in MST configuration.
m_aclk_stream2	Input	Video pipe clock for stream 2. Used in MST configuration.
m_aresetn_stream2	Input	Active-Low video pipe reset for stream 2. Used in MST configuration.
tx_vid_clk_stream*n*	Input	User data clock for MST stream *n*.
tx_vid_rst_stream*n*	Input	Active-High user video reset.
User Ports		
tx_vid_clk_stream1	Input	User video clock.
tx_vid_rst_stream1	Input	User video reset. Active-High.
tx_hpd	Input	Hot-plug detect signal to TX from RX.
Audio Streaming Interface		
s_axis_audio_ingress_aclk	Input	AXI4-Stream clock.
s_axis_audio_ingress_aresetn	Input	Active-Low reset.
s_axis_audio_ingress_tdata[31:0]	Input	Streaming data input. • [3:0] - Preamble Code ◦ 4'b0001: Subframe1/ Start of audio block ◦ 4'b0010: Subframe 1 ◦ 4'b0011: Subframe 2 • [27:4] - Audio Sample Word • [28] - Validity Bit (V) • [29] - User Bit (U) • [30] - Channel Status (C) • [31] - Parity (P)
s_axis_audio_ingress_tid[7:0]	Input	• [3:0] - Audio Channel ID • [7:4] - Audio Packet Stream ID

Table 2-1: **DisplayPort TX Subsystem Ports *(Cont'd)***

Signal Name	Direction from Core	Description
s_axis_audio_ingress_tvalid	Input	Valid indicator for audio data from master.
s_axis_audio_ingress_tready	Output	Ready indicator from DisplayPort source.
External Video PHY Sideband status Interface		
s_axis_phy_tx_sb_status_tdata[7:0]	Output	Sideband status to Video PHY
s_axis_phy_tx_sb_status_tready	Input	Sideband status ready input from Video PHY
s_axis_phy_tx_sb_status_tvalid	Output	Sideband status data valid to Video PHY
External Video PHY clock Interface		
tx_lnk_clk	Input	Link clock input from external Video PHY
External Video PHY Lane n [n = 0 to Lane_Count-1] Interface		
m_axis_lnk_tx_lanen_tdata[31:0]	Output	Lanen Data to External Video PHY
m_axis_lnk_tx_lanen_tvalid	Output	Lanen Data Valid to External Video PHY
m_axis_lnk_tx_lanen_tready	Input	Lanen Data Ready from External Video PHY
m_axis_lnk_tx_lanen_tuser[11:0]	Output	Lanen User data out to External Video PHY
HDCP Key Interface		
hdcp_key_aclk	Input	Key clock
hdcp_key_aresetn	Input	Key Interface reset. Active low
hdcp_key_tdata[63:0]	Input	AXI4-Stream Key Tdata
hdcp_key_last	Input	AXI4-Stream Key Tlast
hdcp_key_tready	Output	AXI4-Stream Key Tready
hdcp_key_tuser[7:0]	Input	AXI4-Stream Key TUSER. KMB should send the Key number from 0 to 41. 0 corresponds to KSV and 1 to 40 are the HDCP Keys count.
hdcp_key_tvalid	Input	AXI4-Stream Key TValid
reg_key_sel[2:0]	Output	To select the one of the eight sets of 40 keys.
start_key_transmit	Output	An Active-High pulse that is used to start key transmit.
Aux Signals		
aux_tx_io_n	Output	Negative polarity AUX Manchester-II data.
aux_tx_io_p	Output	Positive polarity AUX Manchester-II data.
aux_tx_channel_in_p	Input	Positive polarity AUX channel input. Valid when AUX IO Type is unidirectional

Table 2-1: **DisplayPort TX Subsystem Ports *(Cont'd)***

Signal Name	Direction from Core	Description
aux_tx_channel_in_n	Input	Negative polarity AUX channel input. Valid when AUX IO Type is unidirectional
aux_tx_channel_out_p	Output	Positive polarity AUX channel Output. Valid when AUX IO Type is unidirectional
aux_tx_channel_out_n	Output	Negative Polarity AUX channel output. Valid when AUX IO Type is unidirectional
aux_tx_data_out	Output	AUX data out. Valid when AUX IO buffer location is external
aux_tx_data_in	Input	AUX data input. Valid when AUX IO buffer location is external
aux_tx_data_en_out_n	Output	AUX data output enable. Active low. Valid only when AUX IO buffer location is external
Interrupt Interface		
dptxss_dp_irq	Output	Displayport TX IP interrupt out
dptxss_hdcp_irq	Output	HDCP IP interrupt out
dptxss_timer_irq	Output	AXI Timer IP interrupt output valid only when HDCP is enabled

Register Space

This section details registers available in the DisplayPort TX Subsystem. The address map is split into following regions:

- Dual Splitter
- VTC 0 (Up to 3 for 4 streams)
- DisplayPort Transmit
- HDCP Controller
- AXI Timer

TIP: *For details about accessing these registers, see Programming Sequence in Chapter 3.*

Dual Splitter Registers

Table 2-2 defines the Dual Splitter registers.

Table 2-2: **Dual Splitter Register Definitions**

Offset	Register	Access	Value	Definition
0x0000	GENR_CONTROL_REG	R/W	0x2	• [0] – Enables the splitter. • [1] – Register update. • [31] - Soft reset bit. If during power on a value of 2 is written to this register, other registers may be programmed. At the end of programming, set this register to 0x3.
0x0008	GENR_ERROR_REG	R/W	0x0	• [0] – Slave EOL early. • [1] – Slave EOL late. • [2] – Slave SOF early. • [3] – Slave SOF late.
0x000C	IRQ_ENABLE	R/W	0	[0] – Interrupt based on the error conditions.
0x0020	TIME_CONTROL REG(1)	R/W	0x0870_0F00	Contains the input image size: • [15:0] - Height. • [15:0] - Width. ***Note:*** For 4k@60 frame split mode, HRES must be programmed to actual HRES/4.
0x0100	CORE_CONTROL_REG	R/W	0x00_01_01_01	For 4k@60 frame split mode, this register can be programmed to 0x020404. For all other modes, it can be 0x10404. • [7:0] – Input number of samples per clock. • [15:8] – Output number of samples per clock. • [23:16] – Number of image segments. • [31:24] – Number of samples overlapping the segments. Should be programmed to 0 because the subsystem supports two frames without overlap.

Notes:

1. Height refers to VRES and the WIDTH refers to HRES.

Video Timing Controller Registers

For details about the Video Timing Controller (VTC) registers, see the *Video Timing Controller Product Guide* (PG016) [Ref 11].

DisplayPort Registers

For details about the DisplayPort registers, see the *DisplayPort Product Guide* (PG064) [Ref 9].

HDCP Registers

For details about the HDCP registers, see the *HDCP Controller Product Guide* (PG224) [Ref 12]

AXI Timer Registers

For details about the AXI Timer registers, see the *AXI Timer Product Guide* (PG079) [Ref 13]

Designing with the Core

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

Clocking

This section describes the link clock (`tx_lnk_clk`) and the video clock (`tx_vid_clk_stream1`). For information on other clocks, see the *DisplayPort Product Guide* (PG064) [Ref 9].

The `tx_vid_clk_stream1` and the AXI4-Stream clock (`s_axis_aclk_stream1`) can be the same or AXI4 stream clock can be at higher frequency and the AXI4-Stream to Video bridge can handle asynchronous clocking. The value is based on the Consumer Electronics Association (CEA)/VESA Display Monitor Timing (DMT) standard for given video resolutions. Similarly for MST mode, `tx_vid_clk_streamn` and `s_axis_aclk_streamn` can be same or the `s_axis_aclk_streamn` can be at higher frequency than `tx_vid_clk_streamn`.

The `tx_lnk_clk` is a link clock input to the DisplayPort TX Subsystem generated by the Video PHY (GT). The frequency of `tx_lnk_clk` is `<line_rate>`/40 Mhz for the 32-bit video PHY(GT) data interface. See Table 3-1 for the recommended values.

Table 3-1: **Clocking**

Resolution	AXI4-Stream (s_axis_aclk_stream1)	Video Pipe (m_aclk_stream1)	User Video Clock (tx_vid_clk_stream1)
4k2k at the 60 fps (frame split mode)	148.5(1)	74.25(1)	74.25(1)
Other Modes	Video Clock(2)	Video Clock(2)	Video Clock(2)

Notes:

1. For MST stream 1 and stream 2 only.
2. For all four streams when MST mode is enabled. See DMT/CEA spec for video clock range for each DMT resolution.

Resets

The subsystem has one reset input for each of the AXI4-Lite, AXI4-Stream and Video interfaces:

- `s_axi_aresetn`: Active-Low AXI4-Lite reset. This resets all the programming registers.
- `tx_vid_reset_stream1`: Active-High video pipe reset. For MST with four streams, there are four video resets.
- `s_axis_aresetn_stream1`: Active-Low AXI4-Stream interface reset. For MST with four streams, there are four resets corresponding to each stream.
- `m_aresetn_stream1`: Active-Low reset for streams one and two.

Address Map Example

Table 3-2 shows an example based on a subsystem base address of `0x44C0_0000` (19 bits). The DisplayPort TX Subsystem requires a 19-bit address mapping, starting at an offset address of `0x00000`.

Table 3-2: **Address Map Example**

	SST	MST
DisplayPort TX Core	0x44C0_0000	0x44C0_0000
Dual Splitter	N/A	0x44C1_0000
VTC 0	0x44C2_0000	0x44C2_0000
VTC 1 (N=2)	N/A	0x44C3_0000
VTC 2 (N=3)	N/A	0x44C4_0000
VTC 3 (N=4)	N/A	0x44C5_0000
HDCP Controller	0x44C3_0000	N/A
AXI Timer	0x44C4_0000	N/A

Programming Sequence

This section contains the programming sequence for the subsystem using UHD@60 in MST mode with two streams. Program and enable the components of the DisplayPort TX Subsystem in the following order. HDCP Controller and AXI Timer address map exist when HDCP is enabled in SST mode.

1. Displayport TX Core

2. Dual Splitter
3. Video Timing Controller

Dual Splitter Programming

Use the following steps to program the Dual Splitter.

1. Write 0x02 in GENR_CONTROL_REG. This begins the programming sequence and the Dual Splitter register update bit is set.
2. Write vertical resolution and horizontal resolution in TIME_CONTROL_REG.
3. The Dual Splitter is used in a configuration where the input frame must be split into two vertical halves. Write the overlap, number of segments, output samples per clock and input samples per clock in CORE_CONTROL_REG.

   For 4k frame split mode, write 0x02_04_04 to register 0x100 (number of segments =2; number of samples per clock at output = 4; number of samples per clock at input = 4).

   For other modes, write 0x010404 (number of segments =1 (bypass); number of output samples =4; number of input samples =4).
4. Write 0x03 in GENR_CONTROL_REG to enable the Dual Splitter for programmed resolutions and splitting functionality.

When programming the Dual Splitter, note the following:

- There should be no overlap of the two segments in a frame.
- Segment 0 of the Dual Splitter is the left frame and Segment 1 is the right frame.
- The timing of two segments of the splitter is independent, but by the start of a new line, both the segments complete the previous line.
- For 4k@60 in frame split mode, the width of the frame (HRES) must be equal to actual HRES/4.

Design Flow Steps

This chapter describes customizing and generating the subsystem. More detailed information about the standard Vivado® design flows and the IP integrator can be found in the following Vivado Design Suite user guides:

- *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 1]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 3]
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 4]

Customizing and Generating the Subsystem

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

If you are customizing and generating the subsystem in the Vivado IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 1] for detailed information. IP integrator might auto-compute certain configuration values when validating or generating the design. To check whether the values do change, see the description of the parameter in this chapter. To view the parameter value, run the `validate_bd_design` command in the Tcl console.

You can customize the subsystem by specifying values for the various parameters associated with the subsystem IP cores using the following steps:

1. Select the subsystem from the IP catalog.
2. Double-click the selected subsystem 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 2] and the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 3].

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

Customizing the IP

The configuration screen is shown in Figure 4-1.

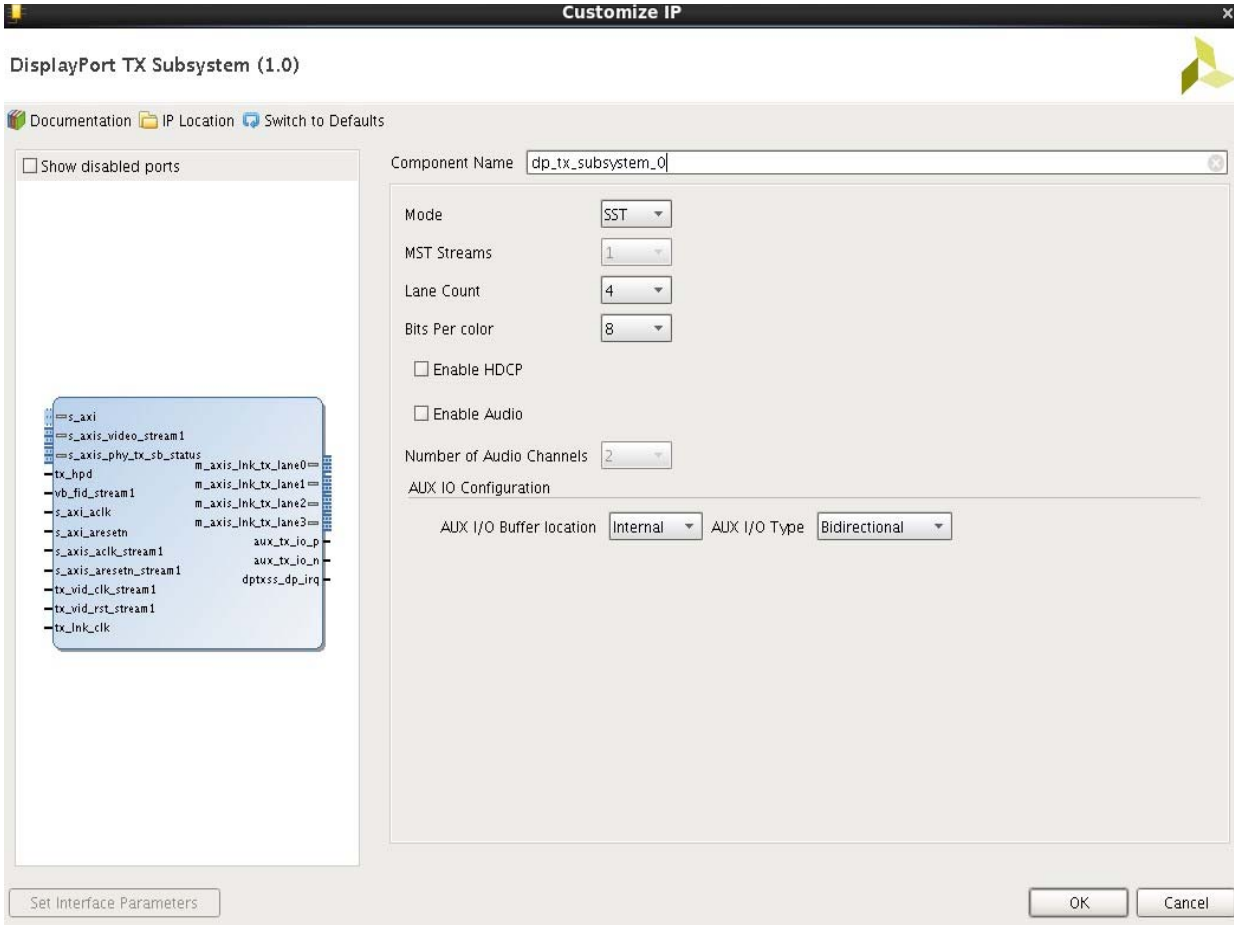

Figure 4-1: **Configuration Screen**

- **Component Name**: The Component Name is used as the name of the top-level wrapper file for the core. The underlying netlist still retains its original name. Names must begin with a letter and must be composed from the following characters: a through z, 0 through 9, and "_". The name displayport_0 is used as internal module name and should not be used for the component name. The default is dp_tx_subsystem_0.
- **Mode**: Select the desired resolution for the video stream out. The default value is SST.
- **MST Streams**: Select the number of streams in MST mode.
- **Lane Count**: Select the number of lanes. Maximum pixel mode supported is aligned with lane count. Pixel mode can be changed dynamically through software but this does not affect the video streaming width.
- **Bits Per Color**: Select the desired bit per color (BPC).
- **Enable Audio**: Enables audio support.

- **Enable HDCP**: Enables HDCP encryption.
- **Audio Channels**: Select the number of audio channels.
- **AUX I/O Buffer location**: Select buffer location for AUX channel
- **AUX I/O Type**: Selection of Bi-Directional or Uni directional buffer type.

User Parameters

Table 4-1 shows the relationship between the GUI fields in the Vivado IDE and the User Parameters (which can be viewed in the Tcl console). The line rate and pixel mode support in the DisplayPort TX Subsystem is through software. Maximum pixel mode support is aligned to the lane count.

Table 4-1: **Vivado IDE Parameter to User Parameter Relationship**

Vivado IDE Parameter/Value	User Parameter/Value	Default Value
Mode	MODE	SST
MST Streams	NUM_STREAMS	1
Lane Count	LANE_COUNT	4
Bits Per Color	BITS_PER_COLOR	8
Enable HDCP	HDCP_ENABLE	0
Enable Audio	AUDIO_ENABLE	0
Number Of Audio Channels	AUDIO_CHANNELS	2
AUX IO Buffer Location	AUX_IO_LOC	Internal
AUX IO Type	AUX_IO_TYPE	Bidirectional

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

Constraining the Core

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

Required Constraints

There are no required constraints for this core.

Device, Package, and Speed Grade Selections

See IP Facts for details about supported devices.

Clock Frequencies

There are no specific clock frequency constraints.

Clock Management

There are no specific clock management constraints.

Clock Placement

There are no specific clock placement constraints.

Banking

There are no specific banking constraints.

Transceiver Placement

There are no specific transceiver placement constraints.

I/O Standard and Placement

There are no specific I/O constraints.

Simulation

There is no example design simulation support for DisplayPort TX Subsystem.

Synthesis and Implementation

For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 2].

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 DisplayPort Subsystem, the Xilinx Support web page (Xilinx Support web page) contains key resources such as product documentation, release notes, answer records, information about known issues, and links for obtaining further product support.

Documentation

This product guide is the main document associated with the DisplayPort Subsystem. 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 Record for the DisplayPort Subsystem

AR: 59384

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 DisplayPort Subsystem 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 6].

Application Software Development

The software is capable of detecting an MST/SST RX connected to the subsystem based on if a MST or SST software flow is executed. Figure B-1 shows the DisplayPort TX Subsystem application software flow.

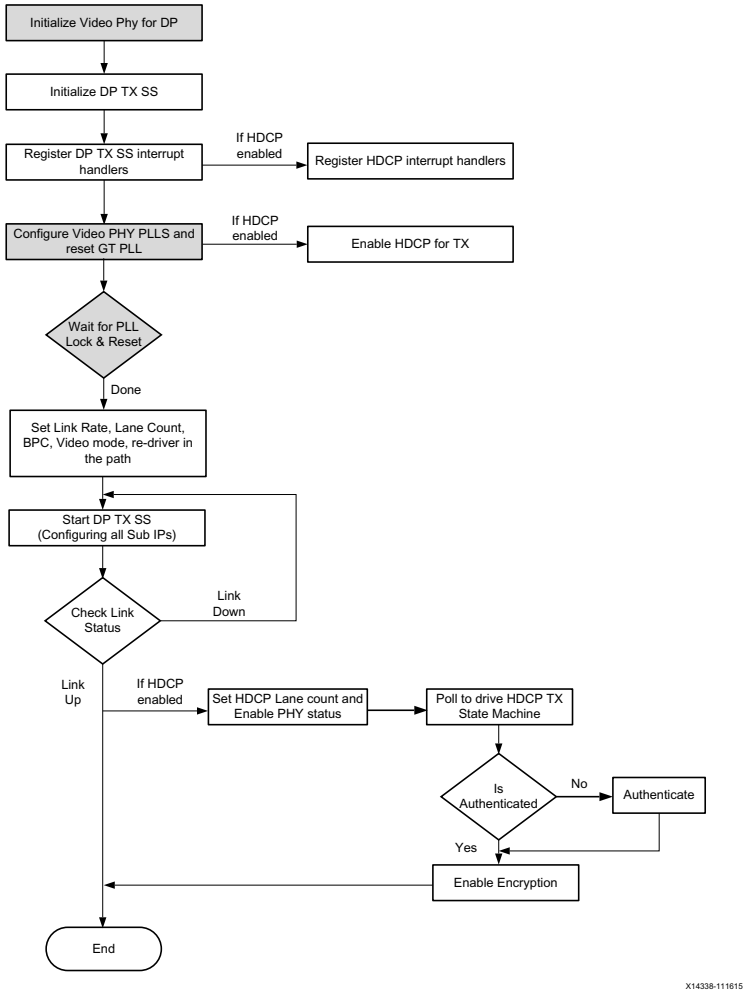

Figure B-1: **Software Flow**

Note: Video PHY is external to DisplayPort TX Subsystem and must be configured for the subsystem to work as expected. For more details on Video PHY configuration, see the *Video PHY Product Guide* (PG230) [Ref 14].

Additional Resources and Legal Notices

Xilinx Resources

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

References

These documents provide supplemental material useful with this product guide:

1. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
2. *Vivado Design Suite User Guide: Designing with IP* (UG896)
3. *Vivado Design Suite User Guide: Getting Started* (UG910)
4. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
5. *ISE to Vivado Design Suite Migration Guide* (UG911)
6. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
7. *Vivado Design Suite User Guide: Implementation* (UG904)
8. *AXI Reference Guide* (UG1037)
9. *DisplayPort Product Guide* (PG064)
10. *AXI4-Stream to Video Out LogiCORE IP Product Guide* (PG044)
11. *Video Timing Controller LogiCORE IP Product Guide* (PG016)
12. *HDCP Controller Product Guide* (PG224)
13. AXI Timer Product Guide (PG079)
14. Video PHY Controller Product Guide (PG230)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
11/18/2015	1.0	Initial Xilinx release.

Please Read: Important Legal Notices