DisplayPort RX Subsystem v1.0

Product Guide

Vivado Design Suite

PG233 November 18, 2015
# Table of Contents

## IP Facts

### Chapter 1: Overview
- Feature Summary .................................................. 5
- Unsupported Features ............................................. 5
- Licensing and Ordering Information .............................. 6

### Chapter 2: Product Specification
- Overview .............................................................. 7
- Standards ............................................................ 11
- Resource Utilization ............................................... 11
- Port Descriptions ................................................... 11
- Register Space ...................................................... 16

### Chapter 3: Designing with the Core
- Clocking ............................................................... 17
- Resets ................................................................. 17
- Programming Sequence ............................................ 18

### Chapter 4: Design Flow Steps
- Customizing and Generating the Subsystem .................... 19
- Constraining the Core ............................................. 21
- Simulation ............................................................ 22
- Synthesis and Implementation .................................... 22

### Appendix A: Debugging
- Finding Help on Xilinx.com ....................................... 23
- Debug Tools .......................................................... 24

### Appendix B: Application Software Development

### Appendix C: Additional Resources and Legal Notices
- Xilinx Resources ..................................................... 26
- References ........................................................... 26
Introduction

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

Features

- Support for DisplayPort Sink (RX) capabilities
- Supports single stream transport (SST) and multi-stream transport (MST)
- Dynamic support for 1.62/2.7/5.4Gbps line rates
- Dynamic support of 6, 8, 10, 12, or 16 bits per color (BPC).
- Dynamic support of RGB and YCbCr444/ YCbCr422/Y-Only color formats.
- Supports Audio
- Supports HDCP
- AXI IIC controller for DP159 retimer programming
- AXI Streaming interface support for video and audio
- Supports 32-bit Video PHY(GT) Interface

LogiCORE IP Facts Table

<table>
<thead>
<tr>
<th>Core Specifics</th>
<th>Supported Device Family(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UltraScale™ Families, UltraScale™ Architecture, Zynq®-7000, 7 Series</td>
</tr>
</tbody>
</table>

| Supported User Interfaces | AXI4-Stream, AXI4-Lite |

| Resources | Performance and Resource Utilization web page |

Provided with Core

| Design Files | Hierarchical subsystem packaged with DisplayPort RX core and other IP cores |
| 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 |

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 RX subsystem is a full feature, hierarchically packaged subsystem with a DisplayPort sink (RX) core ready to use in applications in large video systems.

Feature Summary

• Supports multi-stream transport (MST) and Single Stream Transport (SST) modes of DisplayPort.
• Dynamic support of different BPC and color formats.
• Pixel mode aligns with line count during generation subsystem and can be controlled through software.
• Support for 2 to 8 channel audio.
• 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 Receive core, which is provided under the terms of the Xilinx Core License Agreement. For full access to all core functionalities 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 RX Subsystem operates in the following video modes:

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

In the SST mode, by default the subsystem is packaged with three mandatory subcores: DisplayPort Receive core, Video to AXI4-Stream Bridge and AXI IIC controller. HDCP core along with an AXI Timer core is also present as part of the DisplayPort RX Subsystem, when the HDCP feature is enabled.

In the MST mode, in addition to the subcores listed in SST, Video to AXI4-Stream Bridge instances increase to the number of video streams.

Because the DisplayPort RX 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 DisplayPort RX Subsystem receives the video using the DisplayPort v1.2 protocol over 32-bit video PHY interface. The DisplayPort RX Subsystem works in conjunction with the Video PHY Controller configured for DP protocol. The subsystem outputs multi-pixel Video to AXI4-Stream Protocol interface.
DisplayPort Receive (RX)

The DisplayPort Receive (RX) core contains the following four major blocks as shown in Figure 2-2:

- **Main Link**: Provides for the delivery of the primary video stream.
- **Secondary Channel**: Provides the delivery of audio information from the blanking period of the video stream to an AXI4-Stream interface.
- **AUX Channel**: Establishes the dedicated source to sink communication channel.
- **DPCD**: Contains the set of DisplayPort Configuration Data, which is used to establish the operating parameters of each core.

For more details, see the DisplayPort Product Guide (PG064) [Ref 9].
Chapter 2: Product Specification

Video to AXI4 Stream Bridge

Video to AXI4 stream Bridge is used in DisplayPort RX Subsystem to convert the video output of DisplayPort receive IP to the AXI4 stream standard.

In MST mode, there are N number of bridges in the subsystem, where N = the number of AXI4-Stream outputs to the subsystem. For more details on video pixel mapping over Streaming interface, see the AXI4-Stream Video IP and System Design Guide (UG934) [Ref 10].

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

AXI Interconnect

The subsystem uses Xilinx AXI Interconnect IP core, as a crossbar, which contains one AXI4-Lite slave interface and two AXI4-Lite master interfaces, without HDCP enabled, in the system. With HDCP, the subsystem has four AXI4-Lite master interfaces.

Figure 2-3 shows the AXI slave structure within the DisplayPort RX Subsystem. For more details on the AXI crossbar functionality, see the AXI Interconnect Product Guide (PG059) [Ref 15].
Chapter 2: Product Specification

AXI IIC

AXI IIC controller is used in DisplayPort RX Subsystem to configure the DP159 retimer through IIC interface. The AXI IIC controller in DisplayPort RX Subsystem is working at 400 KHz. DisplayPort RX Subsystem drivers handles the DP159 programming.

For more details on IIC programming for DP159, see the DisplayPort Product Guide (PG064) [Ref 9].

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 decryption along with DisplayPort Receive IP in the DisplayPort RX Subsystem.

Figure 2-4 shows the DisplayPort RX Subsystem with HDCP controller. For more details on HDCP, see the HDCP Product Guide (PG224) [Ref 11].
AXI Timer

A 32-bit AXI Timer is used in DisplayPort RX Subsystem when the HDCP controller is enabled for decryption. The AXI Timer can be accessed through AXI4 master interface for basic timer functionality in the system.

Standards

The DisplayPort RX Subsystem is compatible with the DisplayPort v1.2 Standard, HDCP v1.3 standard, IIC, 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 RX Subsystem ports are described in [Table 2-1](#).
### Table 2-1: DisplayPort RX Subsystem Ports

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Direction from Core</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AXI4-Lite Interface</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_axi_aclk</td>
<td>Input</td>
<td>AXI Bus clock</td>
</tr>
<tr>
<td>s_axi_aresen</td>
<td>Input</td>
<td>AXI reset. Active-Low.</td>
</tr>
<tr>
<td>s_axi_awaddr[13:0]</td>
<td>Input</td>
<td>Write address</td>
</tr>
<tr>
<td>s_axi_awprot[2:0]</td>
<td>Input</td>
<td>Protection Type</td>
</tr>
<tr>
<td>s_axi_awvalid</td>
<td>Input</td>
<td>Write address Valid</td>
</tr>
<tr>
<td>s_axi_awready</td>
<td>Output</td>
<td>Write address Ready</td>
</tr>
<tr>
<td>s_axi_wdata[31:0]</td>
<td>Input</td>
<td>Write data</td>
</tr>
<tr>
<td>s_axi_wstrb[3:0]</td>
<td>Input</td>
<td>Write Strobe</td>
</tr>
<tr>
<td>s_axi_wvalid</td>
<td>Input</td>
<td>Write data valid</td>
</tr>
<tr>
<td>s_axi_wready</td>
<td>Output</td>
<td>Write data ready</td>
</tr>
<tr>
<td>s_axi_bresp[1:0]</td>
<td>Output</td>
<td>Write response</td>
</tr>
<tr>
<td>s_axi_bvalid</td>
<td>Output</td>
<td>Write response valid</td>
</tr>
<tr>
<td>s_axi_bready</td>
<td>Input</td>
<td>Write response ready</td>
</tr>
<tr>
<td>s_axi_araddr[13:0]</td>
<td>Input</td>
<td>Read address</td>
</tr>
<tr>
<td>s_axi_arprot[2:0]</td>
<td>Input</td>
<td>Read protection type</td>
</tr>
<tr>
<td>s_axi_arvalid</td>
<td>Input</td>
<td>Read address valid</td>
</tr>
<tr>
<td>s_axi_arready</td>
<td>Output</td>
<td>Read address ready</td>
</tr>
<tr>
<td>s_axi_rdata[31:0]</td>
<td>Output</td>
<td>Read data</td>
</tr>
<tr>
<td>s_axi_rresp[1:0]</td>
<td>Output</td>
<td>Read data response</td>
</tr>
<tr>
<td>s_axi_rvalid</td>
<td>Output</td>
<td>Read data valid</td>
</tr>
<tr>
<td>s_axi_rready</td>
<td>Input</td>
<td>Read data ready</td>
</tr>
<tr>
<td><strong>DP Video PHY Side Band Status</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_axis_phy_rx_sb_status_tdata[15:0]</td>
<td>Input</td>
<td>Video Phy status input</td>
</tr>
<tr>
<td>s_axis_phy_rx_sb_status_tready</td>
<td>Output</td>
<td>Ready to Video Phy for status</td>
</tr>
<tr>
<td>s_axis_phy_rx_sb_status_tvalid</td>
<td>Input</td>
<td>Video Phy status valid</td>
</tr>
<tr>
<td><strong>DP Video Phy Side Band Control</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m_axis_phy_rx_sb_control_tdata[7:0]</td>
<td>Output</td>
<td>Control Output to Video Phy</td>
</tr>
<tr>
<td>m_axis_phy_rx_sb_control_tvalid</td>
<td>Output</td>
<td>Control output valid to video phy</td>
</tr>
<tr>
<td>m_axis_phy_rx_sb_control_tready</td>
<td>Input</td>
<td>Control data ready input</td>
</tr>
<tr>
<td><strong>DP link Clock Interface</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rx_Lnk_clk</td>
<td>Input</td>
<td>Link clock</td>
</tr>
<tr>
<td><strong>DP Video Phy Main Link [Lane0 -Lane3 ]</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_axis_link_rx_lane0_tdata[31:0]</td>
<td>Input</td>
<td>Main link data for lane0</td>
</tr>
</tbody>
</table>
### DisplayPort RX Subsystem Ports (Cont’d)

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Direction from Core</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>s_axis_lnk_rx_lane0_tvalid</td>
<td>Input</td>
<td>Main link data valid for lane0</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane0_tready</td>
<td>Output</td>
<td>Main link data ready for lane0</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane0_tuser[11:0]</td>
<td>Input</td>
<td>Main link user data for lane0</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane1_tdata[31:0]</td>
<td>Input</td>
<td>Main link data for lane1</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane1_tvalid</td>
<td>Input</td>
<td>Main link data valid for lane1</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane1_tready</td>
<td>Output</td>
<td>Main link data ready for lane1</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane1_tuser[11:0]</td>
<td>Input</td>
<td>Main link user data for lane1</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane2_tdata[31:0]</td>
<td>Input</td>
<td>Main link data for lane2</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane2_tvalid</td>
<td>Input</td>
<td>Main link data valid for lane2</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane2_tready</td>
<td>Output</td>
<td>Main link data ready for lane2</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane2_tuser[11:0]</td>
<td>Input</td>
<td>Main link user data for lane2</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane3_tdata[31:0]</td>
<td>Input</td>
<td>Main link data for lane3</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane3_tvalid</td>
<td>Input</td>
<td>Main link data valid for lane3</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane3_tready</td>
<td>Output</td>
<td>Main link data ready for lane3</td>
</tr>
<tr>
<td>s_axis_lnk_rx_lane3_tuser[11:0]</td>
<td>Input</td>
<td>Main link user data for lane3</td>
</tr>
</tbody>
</table>

**DP Receive Video Interface**
- rx_vid_clk: Input, DisplayPort RX video clock
- rx_vid_rst: Input, DisplayPort RX Video reset

**DP Rx SS Video Stream1 Interface**
- m_axis_aclk_stream1: Input, Stream1 Video clock input
- m_axis_video_stream1_tdata[95:0]: Output, Stream1 Video data
- m_axis_video_stream1_tlast: Output, Stream1 Video last data, End of line pixel.
- m_axis_video_stream1_tready: Input, Stream1 Video data read
- m_axis_video_stream1_tuser: Output, Stream1 video user data
- m_axis_video_stream1_tvalid: Output, Stream1 video data valid

**DP Rx SS Video Stream 2 Interface - MST**
- m_axis_aclk_stream2: Input, Stream2 Video clock input
- m_axis_video_stream2_tdata[95:0]: Output, Stream2 Video data
- m_axis_video_stream2_tlast: Output, Stream2 Video last data, End of line pixel.
- m_axis_video_stream2_tready: Input, Stream2 Video data read
- m_axis_video_stream2_tuser: Output, Stream2 video user data
- m_axis_video_stream2_tvalid: Output, Stream2 video data valid
### Table 2-1: DisplayPort RX Subsystem Ports (Cont’d)

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Direction from Core</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>m_axis_aclk_stream3</td>
<td>Input</td>
<td>Stream3 Video clock input</td>
</tr>
<tr>
<td>m_axis_video_stream3_tdata[95:0]</td>
<td>Output</td>
<td>Stream3 Video data</td>
</tr>
<tr>
<td>m_axis_video_stream3_tlast</td>
<td>Output</td>
<td>Stream3 Video last data, End of line pixel.</td>
</tr>
<tr>
<td>m_axis_video_stream3_tready</td>
<td>Input</td>
<td>Stream3 Video data read</td>
</tr>
<tr>
<td>m_axis_video_stream3_tuser</td>
<td>Output</td>
<td>Stream3 video user data</td>
</tr>
<tr>
<td>m_axis_video_stream3_tvalid</td>
<td>Output</td>
<td>Stream3 video data valid</td>
</tr>
<tr>
<td>m_axis_aclk_stream4</td>
<td>Input</td>
<td>Stream4 Video clock input</td>
</tr>
<tr>
<td>m_axis_video_stream4_tdata[95:0]</td>
<td>Output</td>
<td>Stream4 Video data</td>
</tr>
<tr>
<td>m_axis_video_stream4_tlast</td>
<td>Output</td>
<td>Stream4 Video last data, End of line pixel.</td>
</tr>
<tr>
<td>m_axis_video_stream4_tready</td>
<td>Input</td>
<td>Stream4 Video data read</td>
</tr>
<tr>
<td>m_axis_video_stream4_tuser</td>
<td>Output</td>
<td>Stream4 video user data</td>
</tr>
<tr>
<td>m_axis_video_stream4_tvalid</td>
<td>Output</td>
<td>Stream4 video data valid</td>
</tr>
<tr>
<td>aux_rx_io_p</td>
<td>Inout</td>
<td>Bi-directional AUX IO- P</td>
</tr>
<tr>
<td>aux_rx_io_n</td>
<td>inout</td>
<td>Bi-directional AUX IO- n</td>
</tr>
<tr>
<td>aux_rx_channel_in_p</td>
<td>Input</td>
<td>Unidirectional AUX channel in - P</td>
</tr>
<tr>
<td>aux_rx_channel_in_n</td>
<td>Input</td>
<td>Unidirectional AUX channel in - n</td>
</tr>
<tr>
<td>aux_rx_channel_out_p</td>
<td>Output</td>
<td>Unidirectional AUX channel out - P</td>
</tr>
<tr>
<td>aux_rx_channel_out_n</td>
<td>Output</td>
<td>Unidirectional AUX channel out- n</td>
</tr>
<tr>
<td>aux_rx_data_in</td>
<td>Input</td>
<td>External AUX data input</td>
</tr>
<tr>
<td>aux_rx_data_out</td>
<td>Output</td>
<td>External AUX data output</td>
</tr>
<tr>
<td>aux_rx_data_en_out_n</td>
<td>Output</td>
<td>External AUX data enable out. Active-Low.</td>
</tr>
<tr>
<td>rx_hpd</td>
<td>Output</td>
<td>HPD from DisplayPort RX</td>
</tr>
<tr>
<td>edid_iic_sci_i</td>
<td>Input</td>
<td>EDID IIC SCL input</td>
</tr>
<tr>
<td>edid_iic_sci_o</td>
<td>Output</td>
<td>EDID IIC SCL output</td>
</tr>
</tbody>
</table>
### Table 2-1: DisplayPort RX Subsystem Ports (Cont’d)

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Direction from Core</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>edid_iic_sci_t</td>
<td>Output</td>
<td>EDID IIC SCL enable. IIC SCL enable is Active-Low.</td>
</tr>
<tr>
<td>edid_iic_sda_i</td>
<td>Input</td>
<td>EDID IIC SDA input</td>
</tr>
<tr>
<td>edid_iic_sda_o</td>
<td>Output</td>
<td>EDID IIC SDA output</td>
</tr>
<tr>
<td>edid_iic_sda_t</td>
<td>Output</td>
<td>EDID IIC SDA enable. IIC sda enable is Active-Low.</td>
</tr>
<tr>
<td>dp159_iic_sci_i</td>
<td>Input</td>
<td>DP159 IIC SCL input</td>
</tr>
<tr>
<td>dp159_iic_sci_o</td>
<td>Output</td>
<td>DP159 IIC SCL output</td>
</tr>
<tr>
<td>dp159_iic_sci_t</td>
<td>Output</td>
<td>DP159 IIC SCL enable</td>
</tr>
<tr>
<td>dp159_iic_sda_i</td>
<td>Input</td>
<td>DP159 IIC SDA input</td>
</tr>
<tr>
<td>dp159_iic_sda_o</td>
<td>Output</td>
<td>DP159 IIC SDA output</td>
</tr>
<tr>
<td>dp159_iic_sda_t</td>
<td>Output</td>
<td>DP159 IIC SDA enable</td>
</tr>
<tr>
<td>dp159_rst</td>
<td>Output</td>
<td>DP159 IIC reset through AXI IIC controller GPIO port0</td>
</tr>
</tbody>
</table>

#### DP159 Interface

- dp159_iic_sci_i: Input: DP159 IIC SCL input
- dp159_iic_sci_o: Output: DP159 IIC SCL output
- dp159_iic_sci_t: Output: DP159 IIC SCL enable
- dp159_iic_sda_i: Input: DP159 IIC SDA input
- dp159_iic_sda_o: Output: DP159 IIC SDA output
- dp159_iic_sda_t: Output: DP159 IIC SDA enable
- dp159_rst: Output: DP159 IIC reset through AXI IIC controller GPIO port0

#### 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: Selects one of the eight sets of 40 keys.
- Start_key_transmit: Output: Active-High pulse starts key transmit.

#### Interrupts

- dprxss_dp_irq: Output: DisplayPort RX IP interrupt out
- dprxss_iic_irq: Output: AXI IIC IP interrupt out
- dprx_hdcp_irq: Output: HDCP IP interrupt out
- dprx_timer_irq: Output: AXI Timer Interrupt out
Register Space

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

- DisplayPort Receive (RX) IP
- AXI IIC
- HDCP
- AXI Timer

DisplayPort Registers

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

AXI IIC Registers

For details about the AXI IIC registers, see the AXI IIC Product Guide (PG090) [Ref 13].

HDCP Registers

For details about the HDCP registers, see the HDCP Product Guide (PG224) [Ref 11].

AXI Timer Registers

For details about the AXI Timer registers, see the AXI Timer Product Guide (PG079) [Ref 12].
Chapter 3

Designing with the Core

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

Clocking

This section describes the link clock \(r_{x\_lnk\_clk}\), video clock \(r_{x\_vid\_clk}\) and video bridge AXI4-Stream master interface clock. In the MST mode, single \(r_{x\_vid\_clk}\) connects to all the stream video interfaces. For information on other clocks, see the DisplayPort Product Guide (PG064) [Ref 9].

The \(r_{x\_vid\_clk}\) and the AXI4-Stream master clock can be the same or the AXI4 stream clock can even be higher than video clock frequency \(r_{x\_vid\_clk}\). \(r_{x\_vid\_clk}\) works at 150 Mhz or higher frequencies. Similarly for MST mode, \(r_{x\_vid\_clk}\) and \(m_{axis\_aclk\_streamn}\) can be same or the AXI4 stream clock can even be higher than video clock frequency \(r_{x\_vid\_clk}\).

The \(r_{x\_lnk\_clk}\) is a link clock input to the DisplayPort RX Subsystem generated by the Video PHY (GT). The frequency of \(r_{x\_lnk\_clk}\) is \(<\text{line_rate}>/40\) Mhz for the 32-bit video PHY(GT) data interface.

Resets

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

- \(s_{\text{axi\_aresetn}}\): Active-Low AXI4-Lite reset. This resets all the programming registers.
- \(r_{x\_vid\_rst}\): Active-High video pipe reset. For MST with four streams, there are four video resets.
- \(dp159\_rst\): Active-High soft reset to the DP159 retimer generated through AXI IIC GPIO port. This reset is asserted through AXI IIC register programming for GPIO ports. For more details, see the AXI IIC Controller Product Guide (PG090) [Ref 13].
**Address Map Example**

*Table 3-1* shows an example based on a subsystem base address of 0x44C0_0000 (14 bits). There are no registers in Video to AXI4 Stream bridge.

*Table 3-1: Address Map Example*

<table>
<thead>
<tr>
<th></th>
<th>SST</th>
<th>MST</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisplayPort RX</td>
<td>0x44C0_0000</td>
<td>0x44C0_0000</td>
</tr>
<tr>
<td>AXI IIC Controller</td>
<td>0x44C1_0000</td>
<td>0x44C1_0000</td>
</tr>
<tr>
<td>HDCP Controller</td>
<td>0x44C2_0000</td>
<td>0x44C2_0000</td>
</tr>
<tr>
<td>AXI Timer</td>
<td>0x44C3_0000</td>
<td>0x44C3_0000</td>
</tr>
</tbody>
</table>

**Programming Sequence**

For PHY related programming, see the *Video PHY Controller Product Guide* (PG230) [Ref 14].

For programming sequence of SST/MST modes and audio, see the *DisplayPort Product Guide* (PG064) [Ref 9].

For HDCP related programming sequence, see the *HDCP Controller Product Guide* (PG222) [Ref 11].
Chapter 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_rx_subsystem_0.
- **Mode:** Select the desired resolution for the DisplayPort IP. The default value is SST.
- **MST Streams:** Select the number of streams in MST mode.
- **Lane Count:** Select the number of lanes.
- **Bits Per Color:** Select the desired bit per color (BPC).
- **Enable HDCP:** Enables HDCP.
- **Enable Audio:** Enables audio support.
- **Number of 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).
Table 4-1: Vivado IDE Parameter to User Parameter Relationship

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

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. Being a subsystem, all sub-cores generate their own constraints and the same is applied in the subsystem.

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.
Chapter 4: Design Flow Steps

Banking
There are no specific banking constraints.

Transceiver Placement
Transceiver is external to DisplayPort RX Subsystem hence 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 RX 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 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: 65447

**Technical Support**

Xilinx provides technical support in the [Xilinx Support web page](https://www.xilinx.com) 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](https://www.xilinx.com).

---

**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](https://www.xilinx.com) (UG908) [Ref 6].
Appendix B

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 RX Subsystem application software flow for the SST mode.

Figure B-1: DisplayPort RX Subsystem Software flow for SST mode

Note: Video PHY is external to the DisplayPort RX 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].
Appendix C

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:

5. ISE to Vivado Design Suite Migration Guide (UG911)
8. AXI Reference Guide (UG1037)
10. AXI4-Stream Video IP and System Design Guide (UG934)
11. HDCP Controller Product Guide (PG224)
12. AXI Timer Product Guide (PG079)
13. AXI IIC Bus Interface Product Guide (PG090)
15. AXI Interconnect Product Guide (PG059)
Appendix C: Additional Resources and Legal Notices

Revision History

The following table shows the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/18/2015</td>
<td>1.0</td>
<td>Initial Xilinx release.</td>
</tr>
</tbody>
</table>

Please Read: Important Legal Notices

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

© Copyright 2015 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.