Video PHY Controller v2.0

Product Guide

Vivado Design Suite

PG230 July 14, 2017

XILINX
ALL PROGRAMMABLE™

XILINX
ALL PROGRAMMABLE™

Table of Contents

Chapter 1: Overview

Applications 5
Unsupported Features 5
Licensing and Ordering Information 6

Chapter 2: Product Specification

Video PHY 7
Performance 11
Resource Utilization 11
Port Descriptions 12
Sideband Definitions 18
Register Space 20

Chapter 3: Designing with the Core

Clocking 42
Resets 51
Interrupts 51
Program and Interrupt Flow 52
Video PHY Controller HDMI Implementation 57

Chapter 4: Design Flow Steps

Customizing and Generating the Core 68
Constraining the Core 76
Board Design Guidelines 85
Simulation 86
Synthesis and Implementation 87

Chapter 5: Example Design

HDMI VPHY Example Design 88
DisplayPort VPHY Example Design 88

Appendix A: Verification, Compliance, and Interoperability

Simulation 89
Hardware Testing 89

Appendix B: Migrating and Upgrading

Appendix C: Debugging

Finding Help on Xilinx.com 91
Interface Debug 93

Appendix D: Application Software Development

Appendix E: Additional Resources and Legal Notices

Xilinx Resources 96
Documentation Navigator and Design Hubs 96
References 97
Revision History 98
Please Read: Important Legal Notices 99

Introduction

The Xilinx® Video PHY Controller LogiCORE™ IP core is designed for enabling plug-and-play connectivity with Video (DisplayPort and HDMI® technology) MAC Transmit or Receive subsystems(1). The interface between the video MAC and PHY layers are standardized to enable ease of use in accessing shared transceiver resources. The AXI4-Lite register interface is provided to enable dynamic accesses of transceiver controls/status.

Features

- AXI4-Lite support for register accesses
- Protocol Support: DisplayPort, HDMI
- Full transceiver dynamic reconfiguration port (DRP) accesses and transceiver functions
- Independent TX and RX path line rates (device specific)
- Single quad support
- Phase-locked loop (PLL) switching support from software
- Transmit and Receiver user clocking
- Protocol specific functions (For example, HDMI Clock Detector)
- Non-integer data recovery unit (NI-DRU) support for lower line rates. NI-DRU support is for the HDMI protocol only.
- Advanced Clocking Support (DisplayPort protocol only)

1. For HDMI, appropriate HDMI 2.0 cable driver (TX) and EQ/retimer (RX) devices are required to meet HDMI electrical compliance. Video PHY Controller is not compliant with TMDS specification.

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	**HDMI Video PHY Controller** UltraScale+™ Families (GTHE4) (2) UltraScale Families (GTHE3) 7-Series (GTXE2) Artix®-7 (GTPE2) (5)
	DisplayPort Video PHY Controller UltraScale+™ Families (GTHE4) (2) UltraScale Families (GTHE3) 7-Series (GTXE2) 7-Series (GTHE2) (Discontinued in Vivado 2017.3) Artix®-7 (GTPE2) (Discontinued in Vivado 2017.4)
Supported User Interfaces	AXI4-Lite, AXI4-Stream
Resources	Performance and Resource Utilization web page
Provided with Core	
Design Files	Verilog
Example Design	Not Provided
Test Bench	Not Provided
Constraints File	Xilinx Design Constraints (XDC)
Simulation Model	Not Provided
Supported S/W Driver(3)	Standalone
Tested Design Flows(4)	
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. The maximum line rate for DisplayPort is 2.7 Gbps for GTHE4 Zynq UltraScale+ -1LI (0.72V) devices operated at 16-bit or 20-bit internal datapath.
3. Standalone driver details can be found in the software development kit (SDK) directory (<*install_directory*>/doc/usenglish/xilinx_drivers.htm). Linux OS and driver support information is available from the Xilinx Wiki page.
4. For the supported versions of the tools, see the Xilinx Design Tools: Release Notes Guide.
5. GTPE2 -1, -1L, and -2LE (0.9V) parts are not supported by the HDMI Video PHY Controller because the maximum line rate for those devices is 3.75 Gbps.

Overview

This chapter contains an overview of the core as well as details about applications, licensing, and standards. The Video PHY Controller core is a feature-rich soft IP core incorporating all the necessary logic to properly interface with media access control (MAC) layers and perform physical-side interface (PHY) functionality. Xilinx® IP cores have been successfully tested on hardware and verified. For additional details on the interoperability results, contact your local Xilinx sales representative.

Applications

The Video PHY controller core is the supported method of configuring and using the PHY layer with video MAC controllers.

By separating the PHY layer from the controller layer, the Video PHY provides users with the flexibility of sharing GTs between a video interface input and output or between two different video interfaces such as HDMI and Displayport.

Unsupported Features

- Multi MAC controllers support (Complex use cases)
- GTY transceiver
- The Video PHY Controller core does not currently support mixed MAC controller support, that is, HDMI on the input and DisplayPort output and so on. The current Video PHY Controller core supports MAC on both the input and output.
- add more bullets
- The Video PHY Controller core does not support multiple protocols per instance (for example, two HDMIs in one VPHY.)
- The Video PHY Controller does not support standalone usage. It is designed to be used with Xilinx HDMI or DisplayPort MAC Subsystems.

Licensing and Ordering Information

This Xilinx LogiCORE™ IP module is provided under the terms of the Xilinx Core License Agreement. The module is shipped as part of the Vivado® Design Suite. For full access to all core functionalities in simulation and in hardware, you must purchase a license for the core. Contact your local Xilinx sales representative for information on pricing and availability.

For more information, visit the Video PHY Controller product web page.

Information about other Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page. For information on pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

Product Specification

The Video PHY Controller core is the supported method of configuring and using transceivers with MAC subsystems. The core simplifies serial transceiver (GT) usage by providing a standardized interface and software programmability of serial transceiver functions. These concepts, as well as technical specifications, are described in this chapter.

Video PHY

The PHY is intended to simplify the use of serial transceivers and adds domain-specific configurability. The Video PHY Controller IP is not intended to be used as a stand alone IP and must be used with Xilinx Video MACs such as HDMI 1.4/2.0 Transmitter/Receiver Subsystems and DisplayPort TX/RX Subsystems. The core enables simpler connectivity between MAC layers for TX and RX paths. However, it is still important to understand the behavior, usage, and any limitations of the transceivers. See the device specific transceiver user guide for details.

This chapter introduces the Video PHY Layer and architects Video IP Solutions with a clear boundary between the Link Layer and PHY Layer.

Figure 2-1 shows the standard OSI Model and mapping it with Video IP Solutions.

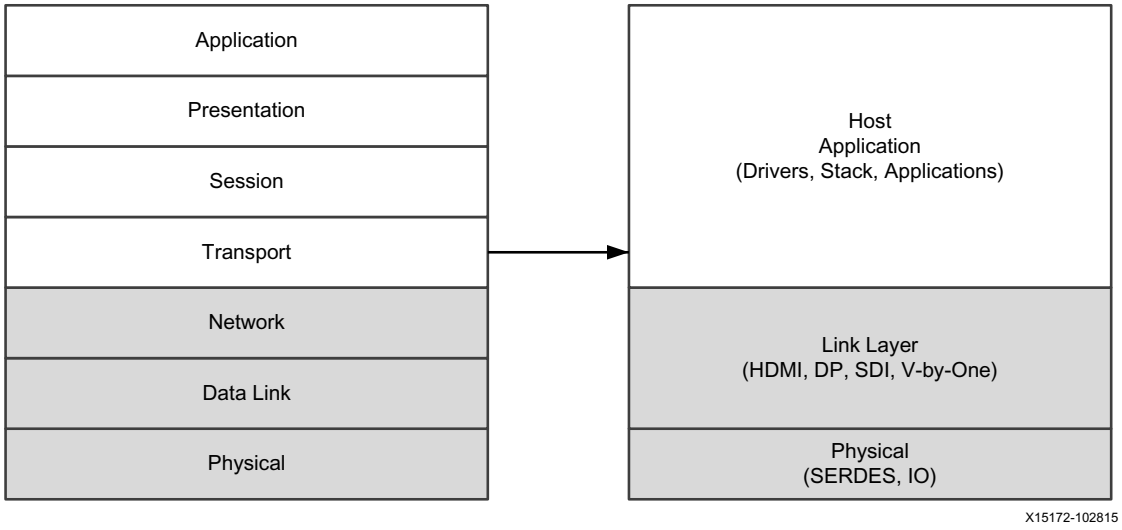

Figure 2-1: **OSI Mapping of Video Systems**

In accordance with the OSI model, the major PHY component for video IP cores is SerDes. Standardizing the SerDes delivery model provides benefits and flexibility for a video MAC layer at the system level.

Figure 2-2 shows the boundary between these MAC and PHY layers and key highlights are:

- AXI4-Lite interface to provide software access
- AXI4-Stream-based GT channel interface for easier connectivity between different video link layers. (GT is also referred to as a serial transceiver.)

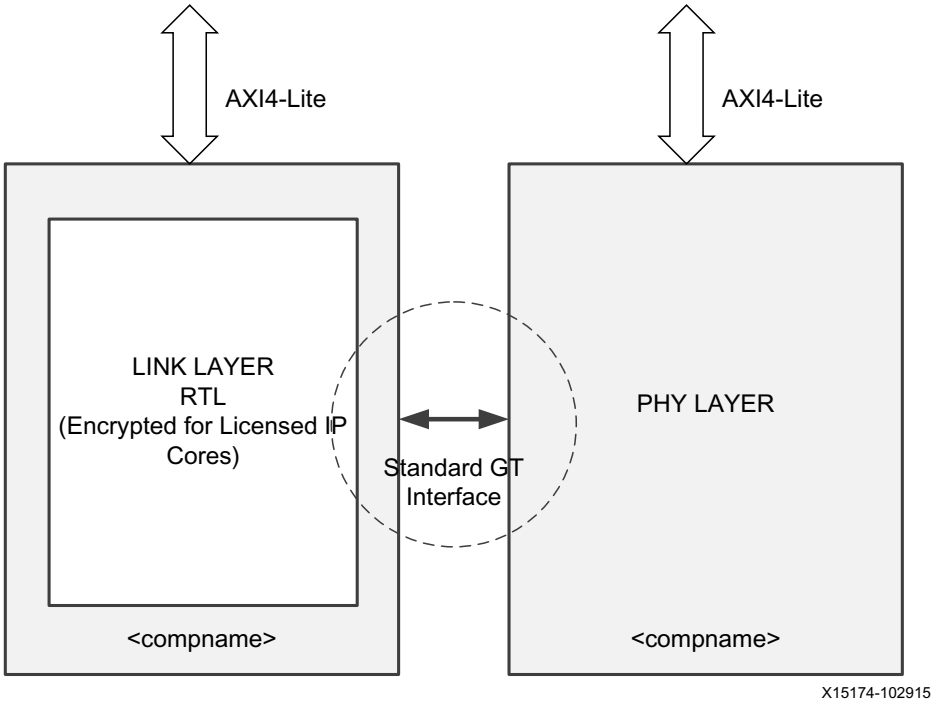

Figure 2-2: **Video IP Layer**

Figure 2-3 shows an overview of the internal structure of the Video PHY Controller core.

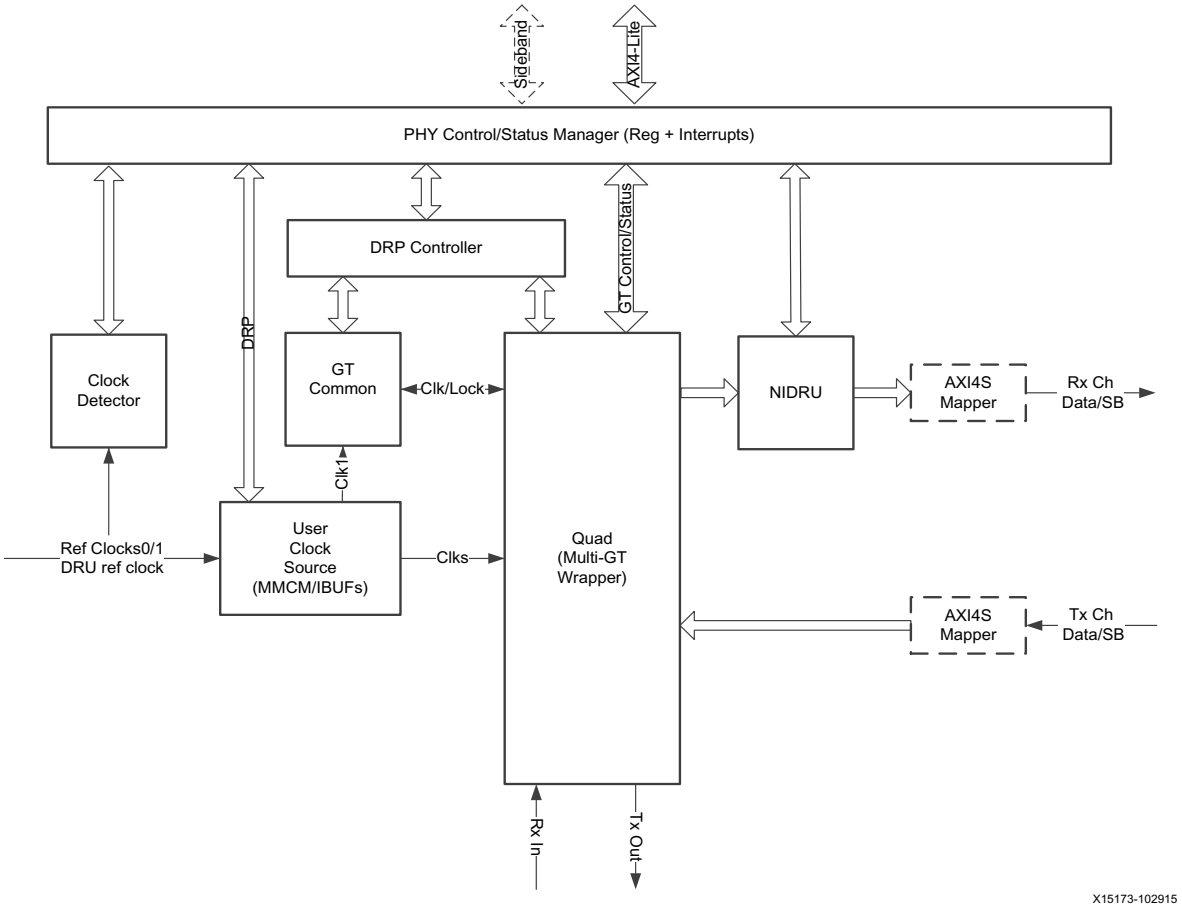

Figure 2-3: **Video PHY Controller Core Block Diagram**

PHY Control/Status Manager

This block manages AXI4-Lite bus protocol accesses and handles memory map accesses and interrupt management.

DRP Controller

This block controls the handshake between AXI4-Lite access and GT DRP access. For example, this block latches `DRP_RDY` and holds it until a read from AXI4-Lite is done. After a proper RDY handshake, a new dynamic reconfiguration port (DRP) transaction can be initiated.

User Clock Source

This block has the GT Input clock buffers and generates USRCLK and USRCLK2 for GTs. In case of TX buffer bypass, Mixed-mode clock manager (MMCM) is part of this module which generates the required output clocks based on TX/RXOUTCLK.

In HDMI, along with generating USRCLK and USRCLK2, this block also produces video clock1 and differential and single-ended TX Transition Minimized Differential Signaling (TMDS) CLK as per requirement of the *HDMI 1.4/2.0 Transmitter Subsystem LogiCORE IP Product Guide* (PG235) [Ref 17] and *HDMI 1.4/2.0 Receiver Subsystem LogiCORE IP Product Guide* (PG236) [Ref 18]. It also buffers the RX TMDS CLK and forwards it as differential and single-ended clocks for generic use.

Note: The video clock maximum frequency is 297 MHz across all transceiver types except GTPE2 which is maxed at 148.5 MHz. This means GTPE2 cannot support video formats with video clocks > 148.5 MHz. For example, it cannot support 4Kp 60Hz at 2 PPC because it requires 297 MHz video clock. 4Kp 60Hz can be supported at 4 PPC because the video clock it requires is only 184.5 MHz. For more information on HDMI clocking requirements, see the "Clocking" sections of *HDMI 1.4/2.0 Transmitter Subsystem LogiCORE IP Product Guide* (PG235) [Ref 17] and *HDMI 1.4/2.0 Receiver Subsystem LogiCORE IP Product Guide* (PG236) [Ref 18].

GT Common

This block controls the COMMON primitive of the serial transceiver. It has the external PLL management and DRP access. This block is available as part of the PHY top level in 7 series devices. For UltraScale™ devices, this block is part of the GT wizard core.

AXI4-Stream Mapper

This block/logic maps the GT input or output data according to the AXI4-Stream protocol defined in the GT specification.

NI-DRU

This block is used in applications where lower line rates (those below the rates supported by the respective GTs) are needed. In HDMI, the NI-DRU is enabled when the RX TMDS clock is below the threshold of the specific GT type.

- GTXE2 and GTPE2 Thresholds:
  - QPLL = 74.125 MHz
  - CPLL = 80.000 MHz
- GTHE3 and GTHE4 Thresholds:
  - QPLL0 = 61.250 MHz
  - CPLL = 100.00 MHz

Note: QPLL1, is not used in NI-DRU mode.

NI-DRU requires an additional fixed reference clock to the GT RX on top of the RX TMDS clock to run the low line rate data recovery. For more information on the reference clock frequency requirement per transceiver type, see Video PHY HDMI Reference Clocks Requirements in Chapter 4.

Performance

The PHY Controller is designed to operate in coordination with the performance characteristics of the transceiver primitives it instantiates. For the DisplayPort protocol, a 2-byte and 4-byte internal datapaths are configured.

The following documents provide information about DC and AC switching characteristics. The frequency ranges specified by these documents must be adhered to for proper transceiver and core operation.

- *Virtex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS893) [Ref 2]
- *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS892) [Ref 3]
- *Kintex-7 FPGAs Data Sheet: DC and AC Switching Characteristics* (DS182) [Ref 4]
- *Virtex-7 T and XT FPGAs Data Sheet: DC and AC Switching Characteristics* (DS183) [Ref 5]
- *Zynq UltraScale+ MPSoC Data Sheet: DC and AC Switching Characteristics* (DS925) [Ref 6]
- *Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics* (DS181) [Ref 7]

Resource Utilization

For full details about performance and resource utilization, visit Performance and Resource Utilization

The maximum clock frequency results were obtained by double-registering input and output ports to reduce dependence on I/O placement. The inner level of registers used a separate clock signal to measure the path from the input registers to the first output register through the core. The results are post-implementation, using tool default settings except for high effort.

The resource usage results do not include the characterization registers and represent the true logic used by the core. LUT counts include SRL16s or SRL32s.

Clock frequency does not take clock jitter into account and should be derated by an amount appropriate to the clock source jitter specification. The maximum achievable clock frequency and the resource counts might also be affected by other tool options, additional logic in the FPGA, using a different version of Xilinx tools, and other factors.

Port Descriptions

Table 2-1 describes the ports and its interface definitions.

Table 2-1: **PHY Controller Ports**

Name	Direction	Width	Clock Domain	Description
Clocking and Reset				
mgtrefclk0_pad_(p/n)_in	Input	1		Available when Advanced Clock Mode is disabled or when GTREFCLK0 is selected as one of the input clock sources in HDMI. User clock module instantiates input buffers. • 7 series: Connects to GTREFCLK0 • UltraScale and UltraScale+: Connects to GTREFCLK00, GTREFCLK01
mgtrefclk1_pad_(p/n)_in	Input	1		Available when Advanced Clock Mode is disabled or when GTREFCLK1 is selected as one of the input clock sources in HDMI. User clock module instantiates input buffers. • 7 series: Connects to GTREFCLK1 • UltraScale and UltraScale+: Connects to GTREFCLK10, GTREFCLK11
mgtrefclk0_in (For DisplayPort)	Input	1		Available when Advanced Clock Mode is enabled. You must instantiate input buffers at the system level. 7 series: Connects to GTREFCLK0
mgtrefclk1_in (For DisplayPort)	Input	1		Available when Advanced Clock Mode is enabled. You must instantiate input buffers at the system level. 7 series: Connects to GTREFCLK1
gtgrefclk_in (For 7 series devices and DisplayPort)	Input	1		Available when Advanced Clock Mode is enabled. You must instantiate input buffers at the system level. Connects to GTGREFCLK0.
gtnorthrefclk(0/1)_in	Input	1		Available when Advanced Clock Mode is enabled or when GTNORTHCLK0/1 is selected as one of the input clock sources. You must instantiate input buffers at the system level. Connects to GTNORTHREFCLK_0/1 ports.

Table 2-1: **PHY Controller Ports *(Cont'd)***

Name	Direction	Width	Clock Domain	Description
gtsouthrefclk(0/1)_in	Input	1		Available when Advanced Clock Mode is enabled or when GTSOUTHREFCLK0/1 is selected as one of the input clock sources. You must instantiate input buffers at the system level. Connects to GTSOUTHREFCLK_0/1 ports.
gtnorthrefclk(0/1)_odiv2_in (For UltraScale and UltraScale+ HDMI)	Input	1		Available when GTNORTHREFCLK0/1 is selected as one of the input clock sources. This must be connected to the ODIV2 output of the gtnorthrefclk0/1_in input buffer. The ODIV2 output must be configured to output a divided-by-1 clock.
gtsouthrefclk(0/1)_ odiv2_in (For UltraScale and UltraScale+ HDMI)	Input	1		Available when GTSOUTHREFCLK0/1 is selected as one of the input clock sources. This must be connected to the ODIV2 output of the gtsouthrefclk0/1_in input buffer. The ODIV2 output must be configured to output a divided-by-1 clock.
gtnorthrefclk00_in (For UltraScale and UltraScale+ devices)	Input	1		Available when Advanced Clock Mode is enabled or when GTNORTHREFCLK0 is selected as one of the input clock sources and QPLL0 is active. You must instantiate input buffers at the system level. Connects to GTNORTHREFCLK00.
gtnorthrefclk01_in (For UltraScale and UltraScale+ devices)	Input	1		Available when Advanced Clock Mode is enabled or when GTNORTHREFCLK0 is selected as one of the input clock sources and QPLL1 is active. You must instantiate input buffers at the system level. Connects to GTNORTHREFCLK01.
gtnorthrefclk10_in (For UltraScale and UltraScale+ devices)	Input	1		Available when Advanced Clock Mode is enabled or when GTNORTHREFCLK1 is selected as one of the input clock sources and QPLL0 is active. You must instantiate input buffers at the system level. Connects to GTNORTHREFCLK10.
gtnorthrefclk11_in (For UltraScale and UltraScale+ devices)	Input	1		Available when Advanced Clock Mode is enabled or when GTNORTHREFCLK1 is selected as one of the input clock sources and QPLL1 is active. You must instantiate input buffers at the system level. Connects to GTNORTHREFCLK11.

Table 2-1: **PHY Controller Ports *(Cont'd)***

Name	Direction	Width	Clock Domain	Description
gtsouthrefclk00_in (For UltraScale and UltraScale+ devices)	Input	1		Available when Advanced Clock Mode is enabled or when GTSOUTHREFCLK0 is selected as one of the input clock sources and QPLL0 is active. You must instantiate input buffers at the system level. Connects to GTSOUTHREFCLK00.
gtsouthrefclk01_in (For UltraScale and UltraScale+ devices)	Input	1		Available when Advanced Clock Mode is enabled or when GTSOUTHREFCLK0 is selected as one of the input clock sources and QPLL1 is active. You must instantiate input buffers at the system level. Connects to GTSOUTHREFCLK01.
gtsouthrefclk10_in (For UltraScale and UltraScale+ devices)	Input	1		Available when Advanced Clock Mode is enabled or when GTSOUTHREFCLK1 is selected as one of the input clock sources and QPLL0 is active. You must instantiate input buffers at the system level. Connects to GTSOUTHREFCLK10.
gtsouthrefclk11_in (For UltraScale and UltraScale+ devices)	Input	1		Available when Advanced Clock Mode is enabled or when GTSOUTHREFCLK1 is selected as one of the input clock sources and QPLL1 is active. You must instantiate input buffers at the system level. Connects to GTSOUTHREFCLK11.
gteastrefclk(0/1)_in (For HDMI and GTPE2 devices)	Input			Available when GTEASTREFCLK0/1 is selected as one of the input clock sources. Connects to GTEASTREFCLK0/1 ports.
gtwestrefclk(0/1)_in (For HDMI and GTPE2 devices)	Input			Available when GTWESTREFCLK0/1 is selected as one of the input clock sources. Connects to GTWESTREFCLK0/1 ports.
drpclk (For UltraScale and UltraScale+ devices)	Input	1		Free running clock that is used to bring up the UltraScale device GT and to clock GT helper blocks.
vid_phy_tx_axi4s_aclk	Input	1		Transmit Interface Clock
vid_phy_tx_axi4s_aresetn	Input	1	TXUSRCLK2	Transmit Interface Reset Unused port. It can be tied HIGH or left unconnected.
vid_phy_rx_axi4s_aclk	Input	1		Receive Interface Clock
vid_phy_rx_axi4s_aresetn	Input	1	RXUSRCLK2	Receive Interface Reset Unused port. It can be tied HIGH or left unconnected.

Table 2-1: **PHY Controller Ports *(Cont'd)***

Name	Direction	Width	Clock Domain	Description
vid_phy_sb_aclk	Input	1		Sideband Interface Clock Connect to AXI4-Lite Clock.
vid_phy_sb_aresetn	Input	1	Sideband	Sideband Interface Reset Unused port. It can be tied HIGH, left unconnected or connected to the ARESETN port of the PROC_SYS_RESET IP under vid_phy_sb_aclk clock domain.
txoutclk	Output	1		Buffered clock sent out for fabric. Available in HDMI when TX is enabled.
rxoutclk	Output	1		Buffered clock sent out for fabric. Available in HDMI when RX is enabled.
vid_phy_axi4lite_aclk	Input	1		AXI Bus Clock
vid_phy_axi4lite_aresetn	Input	1	AXI4-Lite	AXI Reset. Active-Low. Must be connected to ARESETN that is synched to vid_phy_axi4lite_aclk port (i.e. peripheral_aresetn port of Processor System Reset IP)
tx_refclk_rdy (For HDMI)	Input	1		TX Reference Clock ready or lock indicator. Refer to Video PHY HDMI Reference Clocks Requirements in Chapter 4 for details about tx_refclk_rdy port implementation.
tx_tmds_clk (For HDMI)	Output	1		TX TMDS Clock
tx_tmds_clk_p/n (For HDMI)	Output	1		3-state differential TX TMDS Clock output
tx_video_clk (For HDMI)	Output	1		TX Video Clock
txrefclk_ceb (For HDMI)	Output	1		TX external reference clock IBUFDS CEB. Available when TX selects reference clock source from NORTH, SOUTH, EAST, or WEST. Example GTNORTHREFCLK_0/1.
rx_tmds_clk (For HDMI)	Output	1		RX TMDS Clock
rx_tmds_clk_p/n (For HDMI)	Output	1		3-state differential RX TMDS Clock output
rx_video_clk (For HDMI)	Output	1		RX Video Clock

Table 2-1: **PHY Controller Ports *(Cont'd)***

Name	Direction	Width	Clock Domain	Description
rxrefclk_ceb (For HDMI)	Output	1		RX external reference clock IBUFDS CEB. Available when RX selects reference clock source from NORTH, SOUTH, EAST, or WEST. Example GTNORTHREFCLK_0/1.
GT Channels				
phy_rx[p/n]_in	Input	1* Num. channels	Serial	Positive and Negative inputs of the transceiver channel.
phy_tx[p/n]_out	Output	1* Num. channels	Serial	Positive and Negative outputs of the transceiver channel.
vid_phy_tx_axi4s_ch<i>_ tready	Output	1	TXUSRCLK2	AXI4-Stream based tready indicator <i>: Transceiver channel index
vid_phy_tx_axi4s_ch<i>_ tvalid	Input	1	TXUSRCLK2	AXI4-Stream based tvalid indicator. <i>: Transceiver channel index
vid_phy_tx_axi4s_ch<i>_ tdata	Input	TX_DATA_WIDTH (according to the GT user guide)	TXUSRCLK2	AXI4-Stream based tdata bus <i>: Transceiver channel index GT Mapping: TXDATA_IN
vid_phy_tx_axi4s_ch<i>_ tuser	Input	TX_USER_WIDTH For DisplayPort: 12 bits	TXUSRCLK2	AXI4-Stream based tuser bus <i>: Transceiver channel index GT Mapping: {TXCHARDISPVAL, TXCHARDISPMODE, TXCHARISK} In UltraScale and UltraScale+ devices, TXCHARDISPVAL and TXCHARDISPMODE are represented using TXCTLR0 and TXCTRL1.
vid_phy_rx_axi4s_ch<i>_ tready	Input	1	RXUSRCLK2	AXI4-Stream based tready indicator <i>: Transceiver channel index
vid_phy_rx_axi4s_ch<i>_ tvalid	Output	1	RXUSRCLK2	AXI4-Stream based tvalid indicator. <i>: Transceiver channel index
vid_phy_rx_axi4s_ch<i>_ tdata	Output	RX_DATA_WIDTH (according to the GT user guide)	RXUSRCLK2	AXI4-Stream based tdata bus <i>: Transceiver channel index GT Mapping: RXDATAOUT
vid_phy_rx_axi4s_ch<i>_ tuser	Output	RX_USER_WIDTH For DisplayPort: 12 bits	RXUSRCLK2	AXI4-Stream based tuser bus <i>: Transceiver channel index GT Mapping: {RXNOTINTABLE, RXDISPERR, RXCHARISK} In UltraScale and UltraScale+ devices, RXNOTINTABLE, RXDISPERR, RXCHARISK are represented using RXCTRL0, RXCTR1 and RXCTRL3.

Table 2-1: **PHY Controller Ports *(Cont'd)***

Name	Direction	Width	Clock Domain	Description
Sideband Signals (Optional)				
vid_phy_control_sb_tx_tready	Output	1	Sideband	AXI4-Stream based tready indicator
vid_phy_control_sb_tx_tdata	Input	1	Sideband	AXI4-Stream based tdata bus Not used by the DisplayPort Protocol
vid_phy_control_sb_tx_tvalid	Input	1	Sideband	AXI4-Stream based tvalid
vid_phy_status_sb_tx_tready	Input	1	Sideband	AXI4-Stream based tready indicator
vid_phy_status_sb_tx_tdata	Output	8 (DisplayPort) 1 (HDMI)	Sideband	AXI4-Stream based tdata bus For DisplayPort Protocol, refer to Table 2-2. For the HDMI protocol, refer to Table 2-3.
vid_phy_status_sb_tx_tvalid	Output	1	Sideband	AXI4-Stream based tvalid
vid_phy_control_sb_rx_tready	Output	1	Sideband	AXI4-Stream based tready indicator
vid_phy_control_sb_rx_tdata	Input	8	Sideband	AXI4-Stream based tdata bus For DisplayPort Protocol, refer to Table 2-4.
vid_phy_control_sb_rx_tvalid	Input	1	Sideband	AXI4-Stream based tvalid
vid_phy_status_sb_rx_tready	Input	1	Sideband	AXI4-Stream based tready indicator
vid_phy_status_sb_rx_tdata	Output	16 (DisplayPort) 2 (HDMI)	Sideband	AXI4-Stream based tdata bus For DisplayPort Protocol, refer to Table 2-5. For the HDMI protocol, refer to Table 2-6.
vid_phy_status_sb_rx_tvalid	Output	1	Sideband	AXI4-Stream based tvalid
AXI4-Lite Signals				
vid_phy_axi4lite_awaddr	Input	9	AXI4-Lite	Write address
vid_phy_axi4lite_awprot	Input	3	AXI4-Lite	Protection type
vid_phy_axi4lite_awvalid	Input	1	AXI4-Lite	Write address valid
vid_phy_axi4lite_awready	Output	1	AXI4-Lite	Write address ready
vid_phy_axi4lite_awdata	Input	32	AXI4-Lite	Write data bus
vid_phy_axi4lite_awstrb	Input	4	AXI4-Lite	Write strobes
vid_phy_axi4lite_wvalid	Input	1	AXI4-Lite	Write valid
vid_phy_axi4lite_wready	Output	1	AXI4-Lite	Write ready
vid_phy_axi4lite_bresp	Output	2	AXI4-Lite	Write response

Table 2-1: **PHY Controller Ports *(Cont'd)***

Name	Direction	Width	Clock Domain	Description
vid_phy_axi4lite_bvalid	Output	1	AXI4-Lite	Write response valid
vid_phy_axi4lite_bready	Input	1	AXI4-Lite	Response ready
vid_phy_axi4lite_araddr	Input	9	AXI4-Lite	Read address
vid_phy_axi4lite_arprot	Input	3	AXI4-Lite	Protection type
vid_phy_axi4lite_arvalid	Input	1	AXI4-Lite	Read address valid
vid_phy_axi4lite_arready	Output	1	AXI4-Lite	Read address ready
vid_phy_axi4lite_rdata	Output	32	AXI4-Lite	Read data
vid_phy_axi4lite_rresp	Output	2	AXI4-Lite	Read response
vid_phy_axi4lite_rvalid	Output	1	AXI4-Lite	Read valid
vid_phy_axi4lite_rready	Input	1	AXI4-Lite	Read ready
irq	Output	1	AXI4-Lite	Interrupt output

Sideband Definitions

DisplayPort Transmit — Control Path

No control signals are transferred from the DisplayPort Link to DisplayPort PHY Layer.

DisplayPort Transmit — Status Path

The following status is transferred to the Link Layer. The status bits are driven using the AXI4-Lite Clock.

Table 2-2: **DisplayPort Transmit Status Sideband Definition**

Bit Position	Status Details
0	Bank 0, GT Channel 0, TX Reset Done
1	Based on TXSYSCLKSEL[0], CPLL Channel 0/QPLL Lock is transferred
2	Bank 1, GT Channel 1, TX Reset Done
3	Based on TXSYSCLKSEL[0], CPLL Channel 1/QPLL Lock is transferred
4	Bank 0, GT Channel 2, TX Reset Done
5	Based on TXSYSCLKSEL[2], CPLL Channel 2/QPLL Lock is transferred
6	Bank 0, GT Channel 3, TX Reset Done
7	Based on TXSYSCLKSEL[3], CPLL Channel 3/QPLL Lock is transferred

HDMI Transmit — Status Path

The following status is transferred to the Link Layer. The status bits are driven using the AXI4-Lite Clock.

Table 2-3: **HDMI Transmit Status Sideband Definition**

Bit Position	Status Details
0	TX Link Ready. This signal is asserted to indicate that the GT TX initialization is completed (txresetdone).
1	TX Video Ready. This signal is asserted to indicate that the video clock from TX DCM block is stable.

DisplayPort Receive — Control Path

The following control is transferred from the Link Layer. The control bits are driven using the AXI4-Lite Clock.

Table 2-4: **DisplayPort Receive Control Sideband Definition**

Bit Position	Status Details
0	Training Iteration GT Reset. Pulse generated for every access of the DPCD TRAINING_LANE0_SET register which can be used to reset the GT to eliminate buffer errors and bad CDR locks.
1	Start of TP1 Reset. Pulse generated whenever TP1 pattern starts. This can be used to reset the GT for a clean start of training sequence.

DisplayPort Receive — Status Path

The following status is transferred to the Link Layer. The status bits are driven using the AXI4-Lite Clock.

Table 2-5: **Receive Status Sideband Definition**

Bit Position	Status Details
0	Bank 0, GT Channel 0, RX Reset Done
1	Based on RXSYSCLKSEL[0], CPLL Channel 0/QPLL Lock is transferred
2	Bank 0, GT Channel 0, RX Byte Is Aligned output
3	Bank 0, GT Channel 1, RX Reset Done
4	Based on RXSYSCLKSEL[0], CPLL Channel 1/QPLL Lock is transferred
5	Bank 0, GT Channel 1, RX Byte Is Aligned output
6	Bank 0, GT Channel 2, RX Reset Done
7	Based on RXSYSCLKSEL[0], CPLL Channel 2/QPLL Lock is transferred
8	Bank 0, GT Channel 2, RX Byte Is Aligned output

Table 2-5: **Receive Status Sideband Definition *(Cont'd)***

Bit Position	Status Details
9	Bank 0, GT Channel 3, RX Reset Done
10	Based on RXSYSCLKSEL[0], CPLL Channel 3/QPLL Lock is transferred
11	Bank 0, GT Channel 3, RX Byte Is Aligned output

HDMI Receive — Status Path

The following status is transferred to the Link Layer. The status bits are driven using the AXI4-Lite Clock.

Table 2-6: **HDMI Receive Status Sideband Definition**

Bit Position	Status Details
0	RX Link Ready. This signal is asserted to indicate that the GT RX initialization is completed (rxresetdone).
1	RX Video Ready. This signal is asserted to indicate that the video clock from the RX DCM block is stable.

Register Space

The PHY configuration data is implemented as a set of distributed registers that can be read or written from the AXI4-Lite interface. These registers are synchronous to the AXI4-Lite domain.

Any bits not specified in Table 2-7 are considered reserved and return 0 upon read. The power-on reset values of control registers are 0 unless specified in the definition. Only address offsets are listed in Table 2-7 and the base address is configured by the AXI interconnect at the system level.

Table 2-7: **Register Map**

Address	Access	Register Name	Details
0x0000	RO	Version Register (VR)	For video_phy_controller_v2_0, VR is 32'h02_00_00_00. [31:24]: Core major version. [23:16]: Core minor version. [15:12]: Core version revision. [11:8]: Core Patch details. [7:0]: Internal revision.
0x0004			Reserved
0x0008			Reserved
0x000C	RW	Bank Select (BSR)	[4:0]: TX Bank Selection (Default value is 0) [12:8]: RX Bank Selection (Default value is 0)

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
Shared Features and Resets			
0x0010	RW	Reference Clock Selection (RCS)	[3:0]: QPLL (GTXE2/GTHE2) / QPLL0REFCLKSEL (GTHE3/ GTHE4) / PLL0REFCLKSEL (GTPE2) [7:4]: CPLLREFCLKSEL [11:8]: QPLL1REFCLKSEL (For UltraScale/UltraScale+ devices)/PLL1REFCLKSEL (GTPE2) [15:12]: Reserved [19:16]: Reserved [23:20]: Reserved [27:24]: {TXSYSCLKSEL[1:0], RXSYSCLKSEL[1:0]} [31:28]: {TXPLLCLKSEL[1:0], RXPLLCLKSEL[1:0]} – For UltraScale and UltraScale+ devices
0x0014	RW	PLL Reset (PR)	0: CPLLRESET 1: QPLL0RESET 2: QPLL1RESET – For UltraScale and UltraScale+ devices
0x0018	RO	PLL Lock Status (PLS)	0: CPLLLOCK – Ch0 1: CPLLLOCK – Ch1 2: CPLLLOCK – Ch2 3: CPLLLOCK – Ch3 4: QPLL/QPLL0LOCK/PLL0LOCK 5: QPLL1LOCK – For UltraScale/UltraScale+ devices PLL1LOCK

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x001C	RW	TX Initialization (TXI)	Channel 1: • 0: GTTXRESET • 1: TXPMARESET • 2: TXPCSRESET • 3: TXUSERRDY • 6:4: Reserved • 7: PLL_GT_RESET Channel 2: • 8: GTTXRESET • 9: TXPMARESET • 10: TXPCSRESET • 11: TXUSERRDY • 14:12: Reserved • 15: PLL_GT_RESET Channel 3: • 16: GTTXRESET • 17: TXPMARESET • 18: TXPCSRESET • 19: TXUSERRDY • 22:20: Reserved • 23: PLL_GT_RESET Channel 4: • 24: GTTXRESET • 25: TXPMARESET • 26: TXPCSRESET • 27: TXUSERRDY • 30:28: Reserved • 31: PLL_GT_RESET

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x0020	RO	TX Initialization Status (TXIS)	Channel 1: • 0: TXRESETDONE • 1: TXPMARESETDONE(1) • 2: GTPOWERGOOD (For UltraScale and UltraScale+ devices)(2) • 7:3: Reserved Channel 2: • 8: TXRESETDONE • 9: TXPMARESETDONE(1) • 10: GTPOWERGOOD (For UltraScale and UltraScale+ devices)(2) • 15:11: Reserved Channel 3: • 16: TXRESETDONE • 17: TXPMARESETDONE(1) • 18: GTPOWERGOOD (For UltraScale and UltraScale+ devices)(2) • 23:19: Reserved Channel 4: • 24: TXRESETDONE • 25: TXPMARESETDONE(1) • 26: GTPOWERGOOD (For UltraScale and UltraScale+ devices)(2) • 31:27: Reserved

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x0024	RW	RX Initialization (RXI)	Channel 1: • 0: GTRXRESET • 1: RXPMARESET • 2: RXDFELPMRESET(3) • 3: EYESCANRESET(4) • 4: RXPCSRESET(4) • 5: RXBUFRESET • 6: RXUSERRDY(5) • 7: PLL_GT_RESET Channel 2: • 8: GTRXRESET • 9: RXPMARESET • 10: RXDFELPMRESET(3) • 11: EYESCANRESET(4) • 12: RXPCSRESET(4) • 13: RXBUFRESET • 14: RXUSERRDY(5) • 15: PLL_GT_RESET Channel 3: • 16: GTRXRESET • 17: RXPMARESET • 18: RXDFELPMRESET(3) • 19: EYESCANRESET(4) • 20: RXPCSRESET(4) • 21: RXBUFRESET • 22: RXUSERRDY(5) • 23: PLL_GT_RESET Channel 4: • 24: GTRXRESET • 25: RXPMARESET • 26: RXDFELPMRESET(3) • 27: EYESCANRESET(4) • 28: RXPCSRESET(4) • 29: RXBUFRESET • 30: RXUSERRDY(5) • 31: PLL_GT_RESET

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x0028	RO	RX Initialization Status (RXIS)	Channel 1: • 0: RXRESETDONE • 1: RXPMARESETDONE(6) • 2: GTPOWERGOOD (For UltraScale and UltraScale+ devices)(2) • 7:3: Reserved Channel 2: • 8: RXRESETDONE • 9: RXPMARESETDONE(6) • 10: GTPOWERGOOD (For UltraScale and UltraScale+ devices)(2) • 15:11: Reserved Channel 3: • 16: RXRESETDONE • 17: RXPMARESETDONE(6) • 18: GTPOWERGOOD (For UltraScale and UltraScale+ devices)(2) • 23:19: Reserved Channel 4: • 24: RXRESETDONE • 25: RXPMARESETDONE(6) • 26: GTPOWERGOOD (For UltraScale and UltraScale+ devices)(2) • 31:27: Reserved
0x002C	RW	IBUFDS GTxx Control (IBUFDSGTxxCTRL)	This control is used manage CEB pin of IBUFDS_GTE2/3 based on device. 0: GTREFCLK0_CEB 1: GTREFCLK1_CEB Bank based programming. You must change the appropriate Bank number when working with Multi-Quad.

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
Power Down			
0x0030	RW	Power Down Control (PDC)	Channel 1: • 0: CPLLPD(7) • 1: QPLL0PD/PLL0PD • 2: QPLL1PD (For UltraScale and UltraScale+ devices)/ PLL1PD • 4:3: RXPD[1:0] • 6:5: TXPD[1:0] • 7: Reserved Channel 2: • 8: CPLLPD • 9: QPLL0PD(8) • 10: QPLL1PD (For UltraScale and UltraScale+ devices) • 12:11: RXPD[1:0] • 14:13: TXPD[1:0] • 15: Reserved Channel 3: • 16: CPLLPD • 17: QPLL0PD(8) • 18: QPLL1PD (For UltraScale and UltraScale+ devices) • 20:19: RXPD[1:0] • 22:21: TXPD[1:0] • 23: Reserved Channel 4: • 24: CPLLPD • 25: QPLL0PD(8) • 26: QPLL1PD (For UltraScale and UltraScale+ devices) • 28:27: RXPD[1:0] • 30:29: TXPD[1:0] • 31: Reserved
0x0034			Reserved

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
Loopback			
0x0038	RW	Loopback Control (LBC)	Channel 1: • 2:0: LOOPBACK[2:0] • 7:3: Reserved Channel 2: • 10:8: LOOPBACK[2:0] • 15:11: Reserved Channel 3: • 18:16: LOOPBACK[2:0] • 23:19: Reserved Channel 4: • 26:24: LOOPBACK[2:0] • 31:27: Reserved
0x003C			Reserved
Dynamic Reconfiguration Port (DRP)			
0x0040	RW	DRP CONTROL Channel1 (DRPCCH1)	11:0: DRPADDR[8:0] 12: DRPEN 13: DRPWE 14: DRP Reset (For UltraScale and UltraScale+ devices)(8) 15: Reserved 31:16: DRPDI [15:0]
0x0044	RW	DRP CONTROL Channel2 (DRPCCH2)	11:0: DRPADDR[8:0] 12: DRPEN 13: DRPWE 14: DRP Reset (For UltraScale and UltraScale+ devices)(8) 15: Reserved 31:16: DRPDI [15:0]
0x0048	RW	DRP CONTROL Channel3 (DRPCCH3)	11:0: DRPADDR[8:0] 12: DRPEN 13: DRPWE 14: DRP Reset (For UltraScale and UltraScale+ devices)(8) 15: Reserved 31:16: DRPDI [15:0]

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x004C	RW	DRP CONTROL Channel4 (DRPCCH4)	11:0: DRPADDR[8:0] 12: DRPEN 13: DRPWE 14: DRP Reset (For UltraScale and UltraScale+ devices)(8) 15: Reserved 31:16: DRPDI [15:0]
0x0050	RO	DRP STATUS Channel1 (DRPSCH1)	15:0: DRPDO[15:0] – Valid for Read Transfers 16: DRPRDY (Indicates valid transfer) 17: DRPBUSY (Indicated DRP port free) 31:18: Reserved
0x0054	RO	DRP STATUS Channel2 (DRPSCH2)	15:0: DRPDO[15:0] – Valid for Read Transfers 16: DRPRDY (Indicates valid transfer) 17: DRP BUSY (Indicated DRP port free) 31:18: Reserved
0x0058	RO	DRP STATUS Channel3 (DRPSCH3)	15:0: DRPDO[15:0] – Valid for Read Transfers 16: DRPRDY (Indicates valid transfer) 17: DRP BUSY (Indicated DRP port free) 31:18: Reserved
0x005C	RO	DRP STATUS Channel4 (DRPSCH4)	15:0: DRPDO[15:0] – Valid for Read Transfers 16: DRPRDY (Indicates valid transfer) 17: DRP BUSY (Indicated DRP port free) 31:18: Reserved
0x0060	RW	DRP CONTROL Common (DRPCC)	11:0: DRPADDR[8:0] (8 bits for 7 series) 12: DRPEN 13: DRPWE 15:14: Reserved 31:16: DRPDI [15:0]
0x0064	RO	DRP STATUS Common (DRPSC)	15:0: DRPDO[15:0] – Valid for Read Transfers 16: DRPRDY (Indicates valid transfer) 17: DRP BUSY (Indicated DRP port free) 31:18: Reserved
0x0068-0x006C			Reserved

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
Transmitter Functions			
0x0070	RW	TX Control (TXC)	Channel 1: • 0: TX8B10BEN[12] • 1: TXPOLARITY • 4:2: TXPRBSSEL[2:0] • 5: TXPRBSFORCEERR • 7:6 Reserved Channel 2: • 8: TX8B10BEN[12] • 9: TXPOLARITY • 12:10: TXPRBSSEL[2:0] • 13: TXPRBSFORCEERR • 15:14: Reserved Channel 3: • 16: TX8B10BEN[12] • 17: TXPOLARITY • 20:18: TXPRBSSEL[2:0] • 21: TXPRBSFORCEERR • 23:12: Reserved Channel 4: • 24: TX8B10BEN[12] • 25: TXPOLARITY • 28:26: TXPRBSSEL[2:0] • 29: TXPRBSFORCEERR • 31:20: Reserved

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x0074[9]	RW	TX Buffer Bypass Control (TXBBC)	Channel 1: • 0: TXPHDLYRESET • 1: TXPHALIGN • 2: TXPHALIGNEN • 3: TXPHDLYPD • 4: TXPHINIT • 5: TXDLYRESET • 6: TXDLYBYPASS • 7: TXDLYEN Channel 2: • 8: TXPHDLYRESET • 9: TXPHALIGN • 10: TXPHALIGNEN • 11: TXPHDLYPD • 12: TXPHINIT • 13: TXDLYRESET • 14: TXDLYBYPASS • 15: TXDLYEN Channel 3: • 16: TXPHDLYRESET • 17: TXPHALIGN • 18: TXPHALIGNEN • 19: TXPHDLYPD • 20: TXPHINIT • 21: TXDLYRESET • 22: TXDLYBYPASS • 23: TXDLYEN Channel 4: • 24: TXPHDLYRESET • 25: TXPHALIGN • 26: TXPHALIGNEN • 27: TXPHDLYPD • 28: TXPHINIT • 29: TXDLYRESET • 30: TXDLYBYPASS • 31: TXDLYEN

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x0078	RO	TX Status (TXS)	Channel 1: • 0: TXPHALIGNDONE • 1: TXPHINITDONE(8) • 2: TXDLYRESETDONE(8) • 4:3: TXBUFSTATUS[1:0] • 5: TXBUFFBYPASS_ERROR (UltraScale and UltraScale+ devices) • 7:6: Reserved Channel 2: • 8: TXPHALIGNDONE • 9: TXPHINITDONE(8) • 10: TXDLYRESETDONE(8) • 12:11: TXBUFSTATUS[1:0] • 13: TXBUFFBYPASS_ERROR (UltraScale and UltraScale+ devices) • 15:14: Reserved Channel 3: • 16: TXPHALIGNDONE • 17: TXPHINITDONE(8) • 18: TXDLYRESETDONE(8) • 20:19: TXBUFSTATUS[1:0] • 21: TXBUFFBYPASS_ERROR (UltraScale and UltraScale+ devices) • 23:222: Reserved Channel 4: • 24: TXPHALIGNDONE • 25: TXPHINITDONE(8) • 26: TXDLYRESETDONE(8) • 28:27: TXBUFSTATUS[1:0] • 29: TXBUFFBYPASS_ERROR (UltraScale and UltraScale+ devices) • 31:30: Reserved

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x007C	RW	TX DRIVER Control – Channel 1 and 2(TXDC12)	Channel 1: • 3:0: TXDIFFCTRL[3:0](13) • 4: TXELECIDLE(8) • 5: TXINHIBIT • 10:6: TXPOSTCURSOR[4:0] • 15:11: TXPRECURSOR[4:0] Channel 2: • 19:16: TXDIFFCTRL [3:0](13) • 20: TXELECIDLE(8) • 21: TXINHIBIT • 26:22: TXPOSTCURSOR[4:0] • 31:27: TXPRECURSOR[4:0]
0x0080	RW	TX DRIVER Control – Channel 3 and 4 (TXDC34)	Channel 3: • 3:0: TXDIFFCTRL [3:0](13) • 4: TXELECIDLE(8) • 5: TXINHIBIT • 10:6: TXPOSTCURSOR[4:0] • 15:11: TXPRECURSOR[4:0] Channel 4: • 19:16: TXDIFFCTRL [3:0](13) • 20: TXELECIDLE(8) • 21: TXINHIBIT • 26:22: TXPOSTCURSOR[4:0] • 31:27: TXPRECURSOR[4:0]
0x0084	RC	RX Symbol Error Counter – Channel 1 and2	Error counter increments when there is disparity error and 8B/10B symbol error. Counter is cleared after read operation. [15:0]: Channel 1 error counter. [31:0]: Channel 2 error counter.
0x0088	RC	RX Symbol Error Counter – Channel 3 and 4	Error counter increments when there is disparity error and 8B/10B symbol error. Counter is cleared after read operation. [15:0]: Channel 3 error counter. [31:0]: Channel 4 error counter.
0x008C – 0x009C		Reserved	

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
Receiver Functions			
0x0100	RW	RX Control (RXC)	Channel 1: • 0: Reserved • 1: RX8B10BEN(10) • 2: RXPOLARITY • 3: RXPRBSCNTRESET • 6:4: RXPRBSSEL[2:0] • 7: Reserved Channel 2: • 8: Reserved • 9: RX8B10BEN(10) • 10: RXPOLARITY • 11: RXPRBSCNTRESET • 14:12: RXPRBSSEL[2:0] • 15: Reserved Channel 3: • 16: Reserved • 17: RX8B10BEN(10) • 18: RXPOLARITY • 19: RXPRBSCNTRESET • 22:20: RXPRBSSEL[2:0] • 23: Reserved Channel 4: • 24: Reserved • 25: RX8B10BEN(10) • 26: RXPOLARITY • 27: RXPRBSCNTRESET • 30:28: RXPRBSSEL[2:0] • 31: Reserved

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x0104	RO	RX Status (RXS)	Channel 1: • 0: RXCDRLOCK(11) • 3:1: RXBUFSTATUS [2:0] • 4: RXPRBSERR • 7:5: Reserved Channel 2: • 8: RXCDRLOCK(11) • 11:9: RXBUFSTATUS [2:0] • 12: RXPRBSERR • 15:13: Reserved Channel 3: • 16: RXCDRLOCK(11) • 19:17: RXBUFSTATUS [2:0] • 20: RXPRBSERR • 23:21: Reserved Channel 4: • 24: RXCDRLOCK(11) • 27:25: RXBUFSTATUS [2:0] • 28: RXPRBSERR • 31:29: Reserved

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x0108	RW	RX Equalization and CDR	Channel 1: • 0: RXLPMEN • 1: RXCDRHOLD[8] • 2: RXOSOVRDEN[8] • 3: RXLPMLFKLOVRDEN[8] • 4: RXLPMHFOVRDEN[8] • 5:7: Reserved Channel 2: • 8: RXLPMEN • 9: RXCDRHOLD[8] • 10: RXOSOVRDEN[8] • 11: RXLPMLFKLOVRDEN[8] • 12: RXLPMHFOVRDEN[8] • 13:15: Reserved Channel 3: • 16: RXLPMEN • 17: RXCDRHOLD[8] • 18: RXOSOVRDEN[8] • 19: RXLPMLFKLOVRDEN[8] • 20: RXLPMHFOVRDEN[8] • 21:23: Reserved Channel 4: • 24: RXLPMEN • 25: RXCDRHOLD[8] • 26: RXOSOVRDEN[8] • 27: RXLPMLFKLOVRDEN[8] • 28: RXLPMHFOVRDEN[8] • 29:31: Reserved
0x010C[8]	RW	RX TDLOCK VALUE	[31:0]: TDLock value (Protocol Specific). Default value is 50UI.

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
Interrupts			
0x0110	WO	Interrupt Enable Register (IER)	0: TX Reset Done Change Event 1: RX Reset Done Change Event 2: CPLL Lock Change Event 3: QPLL Lock Change Event/PLL0 Lock Change Event (GTPE2) 4: TX Alignment Done Event 5: QPLL1 Lock Change Event/PLL1 Lock Change Event (GTPE2) 6: Clock Detector TX Frequency Change 7: Clock Detector RX Frequency Change 30: Clock Detector TX Debounce Timeout 31: Clock Detector RX Debounce Timeout
0x0114	WO	Interrupt Disable Register (IDR)	0: TX Reset Done Change Event 1: RX Reset Done Change Event 2: CPLL Lock Change Event 3: QPLL/QPLL0 Lock Change Event/PLL0 Lock Change Event (GTPE2) 4: TX Alignment Done Event 5: QPLL1 Lock Change Event/PLL1 Lock Change Event (GTPE2) 6: Clock Detector TX Frequency Change 7: Clock Detector RX Frequency Change 30: Clock Detector TX Debounce Timeout 31: Clock Detector RX Debounce Timeout
0x0118	RO	Interrupt Mask Register (IMR)	0: TX Reset Done Change Event 1: RX Reset Done Change Event 2: CPLL Lock Change Event 3: QPLL/QPLL0 Lock Change Event/PLL0 Lock Change Event (GTPE2) 4: TX Alignment Done Event 5: QPLL1 Lock Change Event/PLL1 Lock Change Event (GTPE2) 6: Clock Detector TX Frequency Change 7: Clock Detector RX Frequency Change 30: Clock Detector TX Debounce Timeout 31: Clock Detector RX Debounce Timeout

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x011C	W1C	Interrupt Status Register (ISR)	Valid interrupt event by AND of IMR and ISR. • 0: TX Reset Done Change Event: Triggers an interrupt whenever the state of reset done changes. SW has to read the TXIS register to know reset done state. • 1: RX Reset Done Change Event Triggers an interrupt whenever the state of reset done changes. SW has to read the RXIS register to know reset done state. • 2: CPLL Lock Change Event Triggers an interrupt whenever the state of CPLL Lock changes. SW has to read the PLS register to know lock state. • 3: QPLL/QPLL0 Lock Change Event//PLL0 Lock Change Event (GTPE2) Triggers an interrupt whenever the state of QPLL/PLL0 Lock changes. SW has to read the PLS register to know lock state. • 4: TX Alignment Done Event • 5: QPLL1 Lock Change Event/PLL1 Lock Change Event (GTPE2) Triggers an interrupt whenever the state of QPLL1/PLL1 Lock changes. SW has to read the PLS register to know lock state. • 6: Clock Detector TX Frequency Change • 7: Clock Detector RX Frequency Change • 30: Clock Detector TX Debounce Timeout • 31: Clock Detector RX Debounce Timeout
TXUSRCLK Clocking			
0x0120	RW	MMCM TXUSRCLK Control/ Status (MMCM_TXUSRCLK_CTRL)	• 0: Configure New Values: Write to this bit after all multiplier and divider registers are programmed. Self-clearing bit. • 1: MMCM Reset • 8: Configuration Successful (Read Only Bit): The status is polled after bit 0 is set. This bit indicates successful MMCM programming. • 9: MMCM Locked Status • 10: MMCM Power Down • 11: MMCM Lock Mask
0x0124	RW	MMCM TXUSRCLK REG1 (MMCM_TXUSRCLK_REG1)	MMCM attributes. For more details, refer to MMCM documentation. [7:0] -> DIVCLK DIVIDE VALUE [15:8] -> CLKFBOUT MULTIPLER VALUE [25:16] -> CLKFBOUT FRACTIONAL VALUE

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x0128	RW	MMCM TXUSRCLK REG2 (MMCM_TXUSRCLK_REG2)	MMCM attributes. For more details, refer to MMCM documentation. [7:0] -> CLKOUT0 DIVIDE VALUE [25:16] -> CLKOUT0 FRACITONAL VALUE Not used in HDMI.
0x012C	RW	MMCM TXUSRCLK REG3 (MMCM_TXUSRCLK_REG3)	MMCM attributes. For more details, refer to the MMCM documentation. [7:0] -> CLKOUT1 DIVIDE VALUE HDMI TX TMDS Clock Source
0x0130	RW	MMCM TXUSRCLK REG4 (MMCM_TXUSRCLK_REG4)	MMCM attributes. For more details, refer to the MMCM documentation. [7:0] -> CLKOUT2 DIVIDE VALUE HDMI TX Video Clock Source
0x0134	RW	BUFGGT TXUSRCLK Control (BUFGGT_TXUSRCLK_CTRL)	BUFG_GT attributes. For more details, refer to BUFG_GT documentation. • [0] -> CLR: Active-High asynchronous clear forcing BUFG_GT output to zero • [3:1] -> DIV: Specifies the value to divide the clock. Divide value is value provided plus 1. For example, setting 3'b000 provides a divide value of 1 and 3'b111 is a divide value of 8
0x0138	RW	MISC TXUSRCLK Control (MISC_TXUSRCLK_CTRL)	0: CLKOUT1 OBUFTDS Output Enable. 0 - disabled; 1 -enabled HDMI TX TMDS Clock Output Enable 1: TX REFCLK CEB [31:2]: Reserved
0x0140	RW	MMCM RXUSRCLK Control/ Status (MMCM_RXUSRCLK_CTR)	• 0: Configure New Values: Write to this bit after all multiplier and divider registers are programmed. • 1: MMCM Reset • 8: Configuration Successful (Read Only Bit): The status is polled after bit 0 is set. This bit indicates successful MMCM programming. • 9: MMCM Locked Status • 10: MMCM Power Down • 11: MMCM Locked Mask
0x0144	RW	MMCM RXUSRCLK REG1 (MMCM_RXUSRCLK_REG1)	MMCM attributes. For more details, refer to MMCM documentation. [7:0] -> DIVCLK DIVIDE VALUE [15:8] -> CLKFBOUT MULTIPLER VALUE [25:16] -> CLKFBOUT FRACTIONAL VALUE

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x0148	RW	MMCM RXUSRCLK REG2 (MMCM_RXUSRCLK_REG2)	MMCM attributes. For more details, refer to MMCM documentation. [7:0] -> CLKOUT0 DIVIDE VALUE [25:16] -> CLKOUT0 FRACITONAL VALUE Not used in HDMI
0x014C	RW	MMCM RXUSRCLK REG3 (MMCM_RXUSRCLK_REG3)	MMCM attributes. For more details, refer to MMCM documentation. [7:0] -> CLKOUT1 DIVIDE VALUE HDMI RX TMDS Clock Source
0x0150	RW	MMCM RXUSRCLK REG4 (MMCM_RXUSRCLK_REG4)	MMCM attributes. For more details, refer to the MMCM documentation. [7:0] -> CLKOUT2 DIVIDE VALUE HDMI RX Video Clock Source
0x0154	RW	BUFGGT RXUSRCLK Control (BUFGGT_RXUSRCLK_CTRL)	BUFG_GT attributes. For more details, refer to the BUFG_GT documentation. [0] -> CLR: Active-High asynchronous clear forcing BUFG_GT output to zero [3:1] -> DIV: Specifies the value to divide the clock. Divide value is value provided plus 1. For example, setting 3'b000 provides a divide value of 1 and 3'b111 is a divide value of 8.
0x0158	RW	MISC RXUSRCLK Control (MISC_RXUSRCLK_CTRL)	0: CLKOUT1 OBUFTDS Output Enables. 0 - disabled; 1 -enabled HDMI RX TMDS Clock Output Enable 1: RX REFCLK CEB [31:2]: Reserved
Clock Detector (HDMI)			
0x0200	RW	Control Register	0: Run: Set this bit to enable clock detector. 1: TX Timer Clear – Self Clearing 2: RX Timer Clear – Self Clearing 3: TX Frequency Reset – Self Clearing when TX Frequency Zero bit asserts 4: RX Frequency Reset – Self Clearing when TX Frequency Zero bit asserts [12:5] Frequency Lock Counter Threshold
0x204	RO	Status Register	0: TX Frequency Zero 1: RX Frequency Zero 2: TX Reference Clock Locked 3: TX Reference Clock Locked Captured
0x0208	RW	Frequency Counter Timeout	[31:0] Clock Frequency in Hertz (Typically system frequency value)
0x020C	RO	Transmitter Frequency	[31:0] Transmitter Clock Frequency in Hertz
0x0210	RO	Receiver Frequency	[31:0] Receiver Clock Frequency in Hertz

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x0214	RW	Transmitter Timer	[31:0] Transmitter Timeout Value at System Clock per tick
0x0218	RW	Receiver Timer	[31:0] Receiver Timeout Value at System Clock per tick
0x021C	RO	DRU Frequency	[31:0] Data Recovery Unit Clock Frequency in Hertz
Data Recovery Unit			
0x0300	RW	Control Register	DRU Control Register Channel 1: 0: Reset → 0 – Reset asserted; 1 – Reset released 1: Enable → 0 – DRU disabled; 1 – DRU enabled [7:2] Reserved Channel 2: 8: Reset → 0 – Reset asserted; 1 – Reset released 9: Enable → 0 – DRU disabled; 1 – DRU enabled [15:10] Reserved Channel 3: 16: Reset → 0 – Reset asserted; 1 – Reset released 17: Enable → 0 – DRU disabled; 1 – DRU enabled [23:18] Reserved Channel 4: 24: Reset → 0 – Reset asserted; 1 – Reset released 25: Enable → 0 – DRU disabled; 1 – DRU enabled [31:26] Reserved
0x0304	RO	Status Register	[0] Channel 1 DRU Active [1] Channel 2 DRU Active [2] Channel 3 DRU Active [3] Channel 4 DRU Active [23:4] Reserved [31:24] DRU Version
0x0308	RW	Center Frequency Low Register – Channel 1	[31:0] Center frequency bits 31:0
0x030C	RW	Center Frequency High Register – Channel 1	[4:0] Center frequency bits 36:32 [31:5] Reserved
0x0310	RW	Gain Register – Channel 1	[4:0] Gain G1 [7:5] Reserved [12:8] Gain G1 P [16:13] Reserved [20:16] Gain G2 [31:21] Reserved
0x0314	RW	Center Frequency Low Register – Channel 2	[31:0] Center frequency bits 31:0
0x0318	RW	Center Frequency High Register – Channel 2	[4:0] Center frequency bits 36:32 [31:5] Reserved

Table 2-7: **Register Map *(Cont'd)***

Address	Access	Register Name	Details
0x031C	RW	Gain Register – Channel 2	[4:0] Gain G1 [7:5] Reserved [12:8] Gain G1 P [16:13] Reserved [20:16] Gain G2 [31:21] Reserved
0x0320	RW	Center Frequency Low Register – Channel 3	[31:0] Center frequency bits 31:0
0x0324	RW	Center Frequency High Register – Channel 3	[4:0] Center frequency bits 36:32 [31:5] Reserved
0x0328	RW	Gain Register – Channel 3	[4:0] Gain G1 [7:5] Reserved [12:8] Gain G1 P [16:13] Reserved [20:16] Gain G2 [31:21] Reserved
0x032C	RW	Center Frequency Low Register – Channel 4	[31:0] Center frequency bits 31:0
0x0330	RW	Center Frequency High Register – Channel 4	[4:0] Center frequency bits 36:32 [31:5] Reserved
0x0334	RW	Gain Register – Channel 4	[4:0] Gain G1 [7:5] Reserved [12:8] Gain G1 P [16:13] Reserved [20:16] Gain G2 [31:21] Reserved

Notes:

1. For the DisplayPort protocol, TXPMARESETDONE is used only for UltraScale and UltraScale+ devices. For 7 series devices. this field is tied to 1.
2. For the DisplayPort protocol, this field is tied to 1.
3. For the DisplayPort protocol, RXDFELPMRESET is an unused field.
4. For the DisplayPort protocol, this register field is unused for UltraScale and UltraScale+ devices.
5. For the DisplayPort protocol, this register field is unused for UltraScale and UltraScale+ devices. The internal GT UltraScale wizard FSM handles this field.
6. For the DisplayPort protocol, RXPMARESETDONE is used only for UltraScale and UltraScale+ devices. For 7 series devices this field is tied to 1.
7. For the DisplayPort protocol, this field is applicable only for 7 series devices. For UltraScale and UltraScale+ devices, CPLLPD is handled by the GT wizard FSMs internally.
8. For the DisplayPort protocol, this register field is unused.
9. For the DisplayPort protocol, the GT Wizards internal FSM handles these status bits. This register is unused.
10. For the DisplayPort protocol, this field is tied to 1 for 7 series devices. For the HDMI protocol, this register field is unused.
11. For the DisplayPort protocol, this status bus is internally monitored by the GT wizards FSM. This field is unused. For HDMI protocol, this field is unused in GTPE2 devices.
12. For the HDMI protocol, this register field is unused.
13. For the HDMI protocol, this register field is unused in UltraScale and UltraScale+ devices.

For more information, refer to the GT user guides listed in the References in Appendix E.

Designing with the Core

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

Clocking

The Video PHY Controller core internally generates GT wrappers by using the GT Wizard. Refer to the *7 Series FPGAs GTX/GTH Transceivers User Guide* (UG476) [Ref 8] and the *UltraScale Architecture GTH Transceivers User Guide* (UG576) [Ref 9] for more information. The Video PHY Controller core provides an Advanced clocking mode where North and South clocks for the Quad are exposed for external connections.

DisplayPort Clocking

IMPORTANT: *Transmit Buffer Bypass must be enabled for DisplayPort PHY Compliance.*

GTXE2/GTHE2/GTHE3 Clocking When Transmit Buffer Bypass is Disabled

Connect the DP159 forwarded clock to any of the Reference Clock input and use a proper reference clock selection through driver APIs.

`txoutclk_out/rxoutclk_out` are connected to the DisplayPort MAC controller.

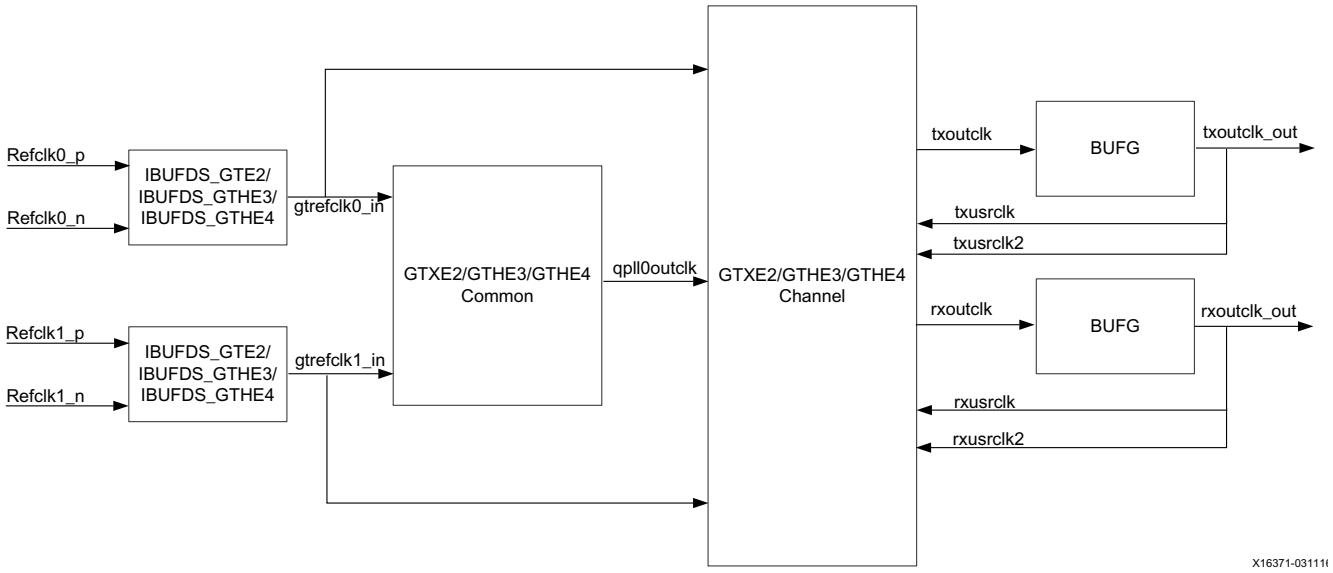

Figure 3-1: **GTXE2/GTHE2/GTHE3 Clocking (TX Buffer Bypass = Disabled)**

GTXE2/GTHE2/GTHE3 Clocking When Transmit Buffer Bypass is Enabled

Connect the DP159 forwarded clock to any of the Reference Clock input and use a proper reference clock selection through driver APIs.

`txoutclk_out/rxoutclk_out` are connected to the DisplayPort MAC controller.

In TX buffer bypass mode, the MMCM clocking resource is used on TX path and you have to program proper divider/multiplier values using driver APIs.

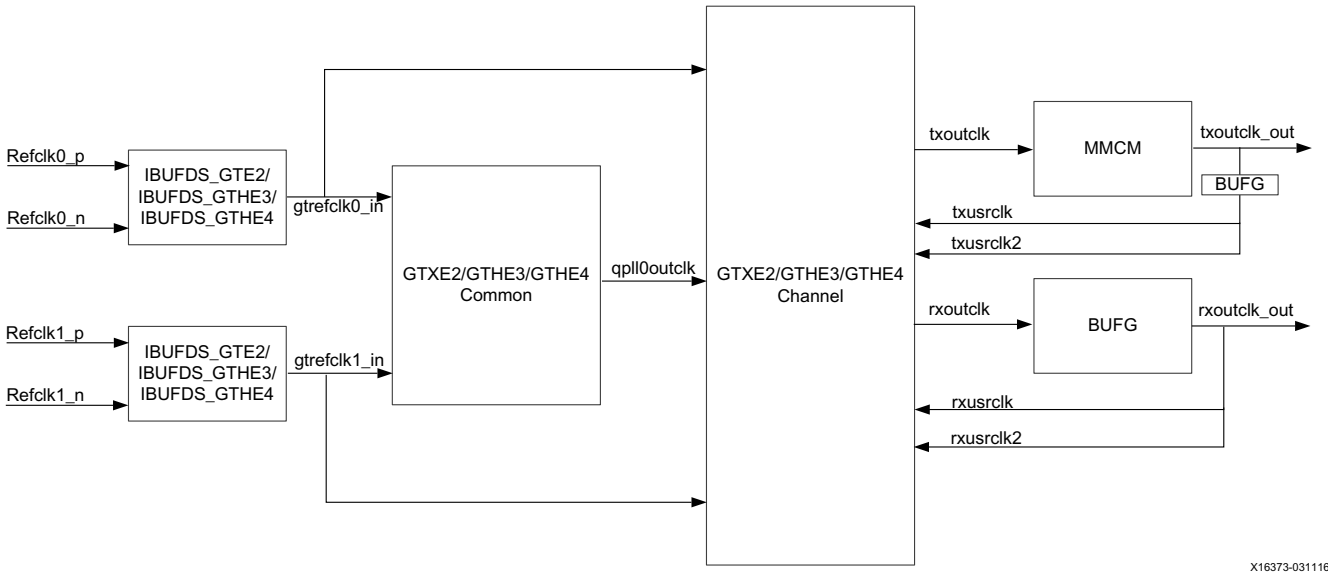

Figure 3-2: **GTXE2/GTHE2/GTHE3 Clocking (TX Buffer Bypass = Enabled)**

GTPE2 Clocking When Transmit Buffer Bypass is Disabled

Connect the DP159 forwarded clock to any of the Reference Clock input and use a proper reference clock selection through driver APIs.

`txoutclk_out/rxoutclk_out` are connected to the DisplayPort MAC controller.

In GTPE2, the MMCM is always used to support 16-bit and 32-bit datapath of the GT channel. MMCM divider and multiplier has to be programmed using driver APIs. `Txusrclk` is connected directly using BUFG.

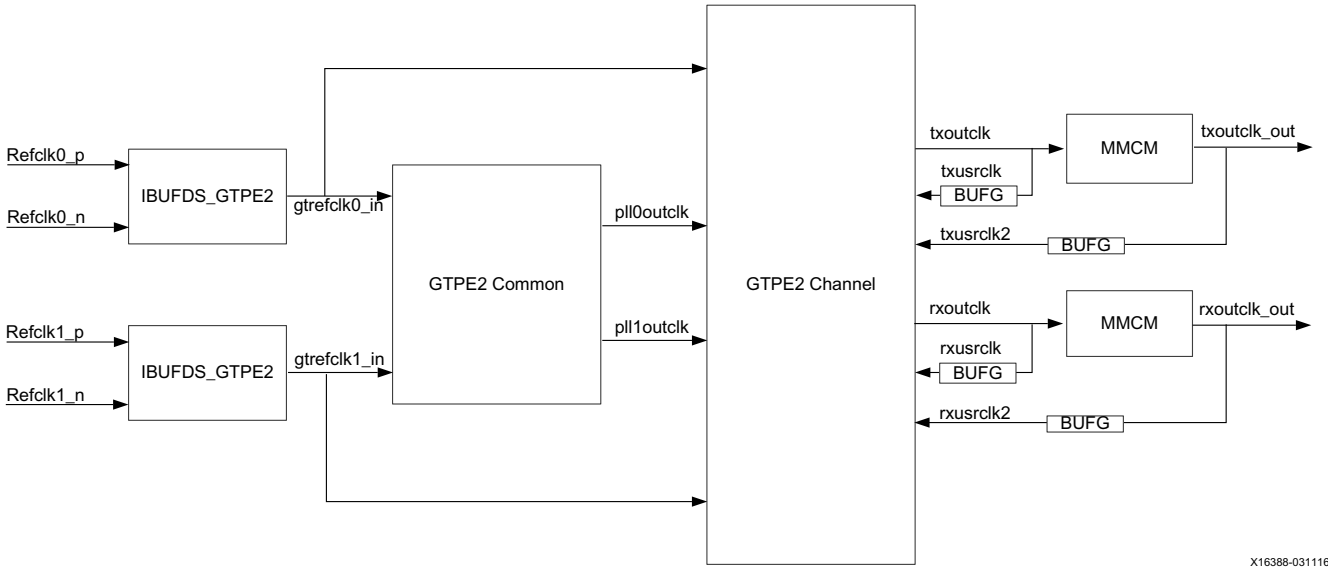

Figure 3-3: **GTPE2 Clocking (TX Buffer Bypass = Disabled)**

GTPE2 Clocking When Transmit Buffer Bypass is Enabled

Connect the DP159 forwarded clock to any of the Reference Clock input and use a proper reference clock selection through driver APIs.

`txoutclk_out/rxoutclk_out` are connected to the DisplayPort MAC controller.

In GTPE2, the MMCM is always used to support 16-bit and 32-bit datapath of the GT channel. MMCM divider and multiplier has to be programmed using driver APIs. `Txusrclk` is connected using MMCM second clock out port.

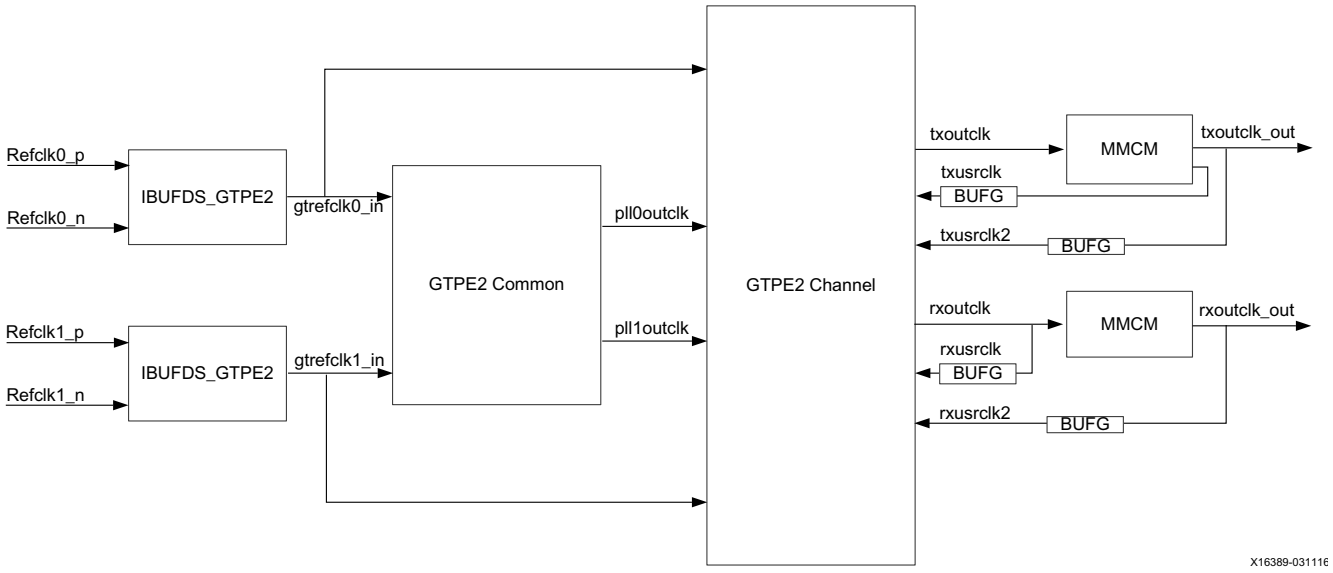

Figure 3-4: **GTPE2 Clocking (TX Buffer Bypass = Enabled)**

HDMI Clocking

IMPORTANT: *The Transmit Buffer Bypass is always enabled for HDMI PHY Compliance.*

The HDMI VPHY clocking diagrams per transceiver type are shown below. Follow the guidelines below when connecting the VPHY clock ports or refer to the HDMI Example Design.

- Connect the external clock generator output clock to the TX reference clock input that was selected in the VPHY GUI. The TX reference clock lock indicator should be connected to the `tx_refclk_rdy` port. See Video PHY HDMI Reference Clocks Requirements in Chapter 4 for its implementation.
- Connect the RX TMDS clock from the external HDMI retimer component clock output to the corresponding RX reference clock input that was selected in the VPHY GUI.
- If NI-DRU is enabled, connect the DRU reference clock to the DRU reference clock input that was selected in the VPHY GUI. Refer to Video PHY HDMI Reference Clocks Requirements in Chapter 4 for the NI-DRU frequency requirements.
- The `txoutclk_out/rxoutclk_out` signal is connected to the HDMI MAC controller.
- The `tx_video_out/rx_video_out` signals are connected to the HDMI MAC controller.
- The `tx_tmds_clk_p/n` signal should be connected to the HDMI TX connector.
- The `tx_tmds_clk` signal can be connected to any logic, e.g. audio generator module
- The `rx_tmds_clk_p/n` signal should be connected to the input of external clock generator if VPHY is used in passthrough mode.

- The `rx_tmds_clk` signal can be connected to any logic.

GTXE2

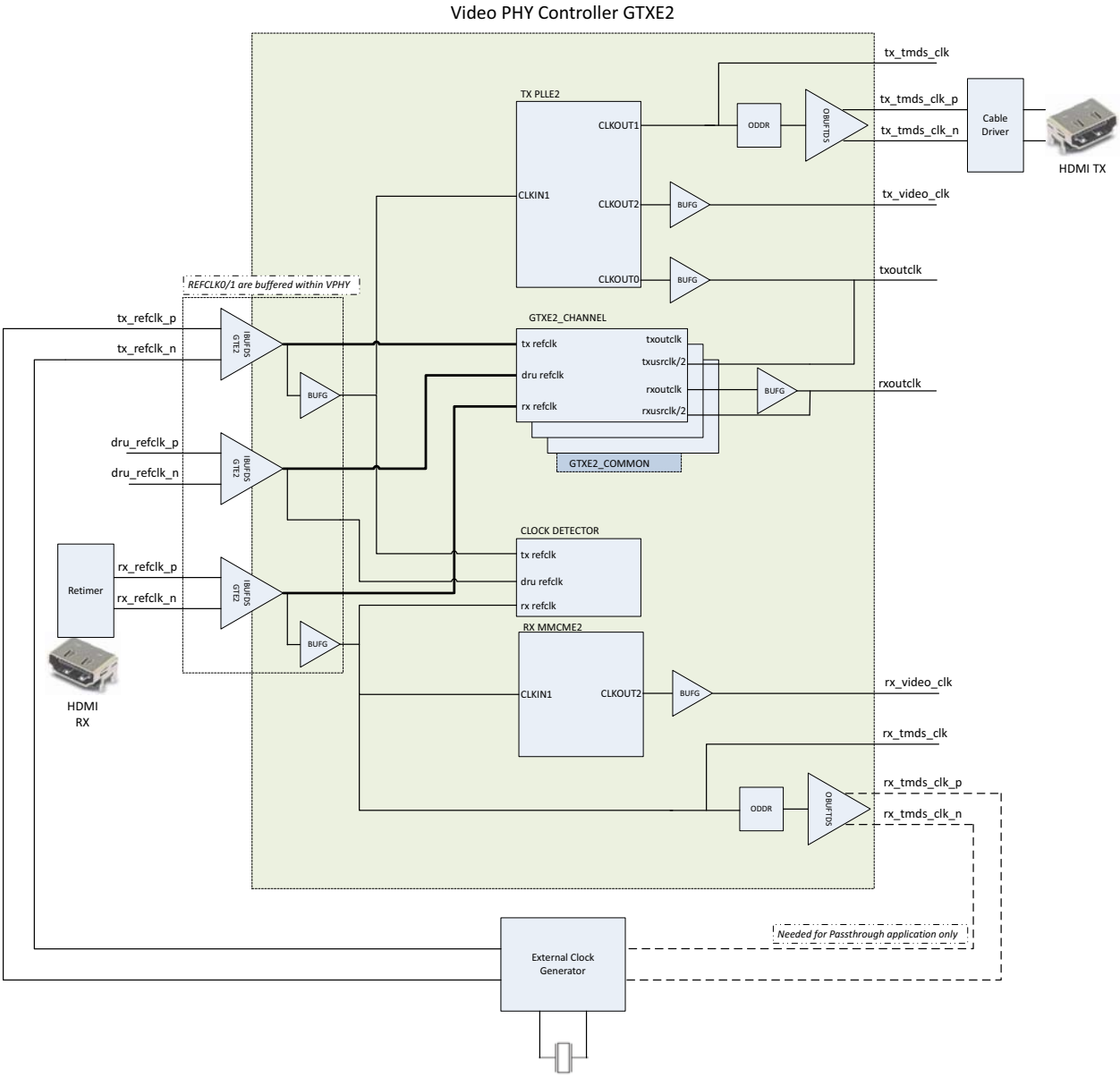

Figure 3-5: **GTXE2**

GTPE2

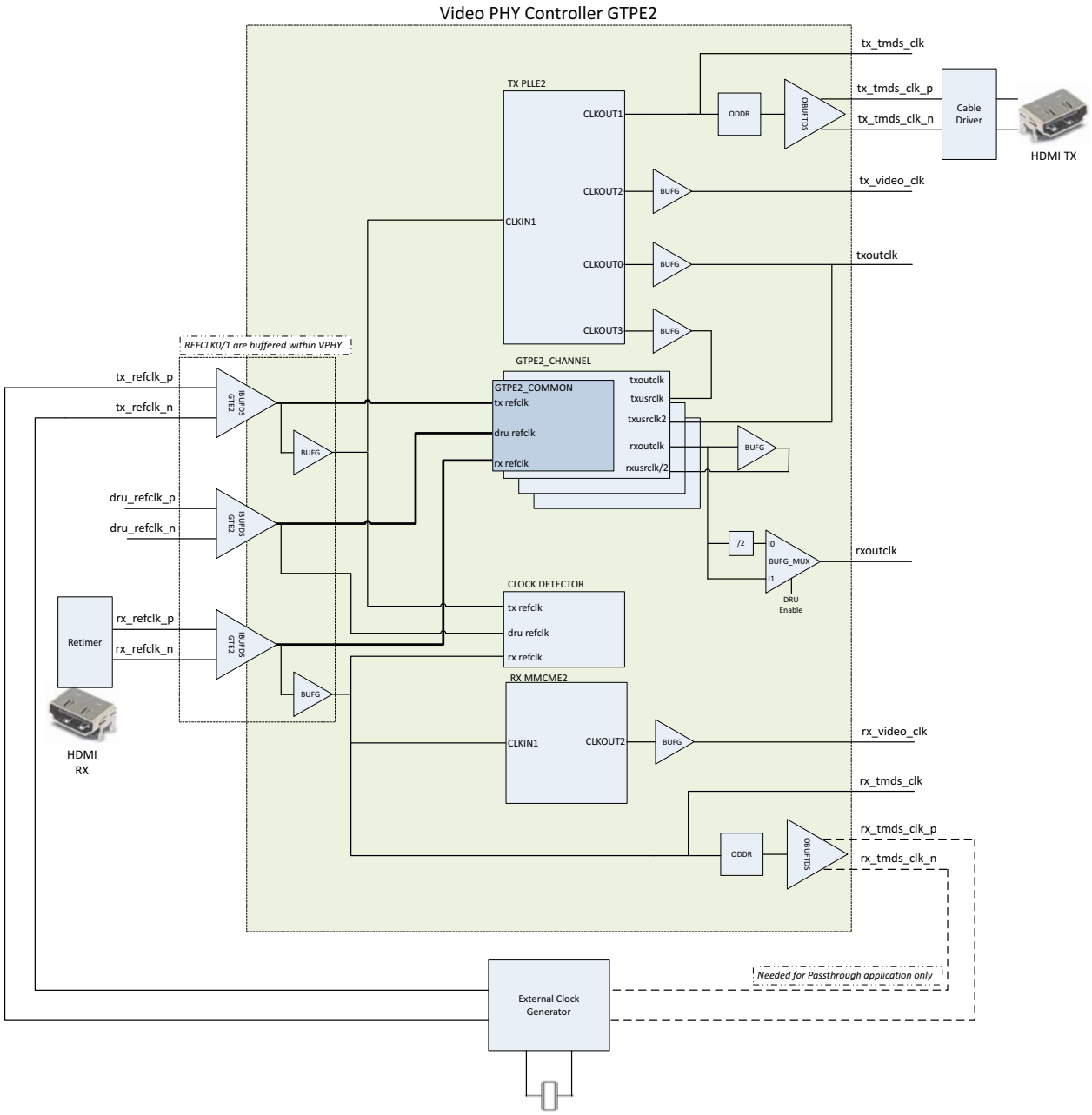

Figure 3-6: **GTPE2**

GTHE3 and GTHE4

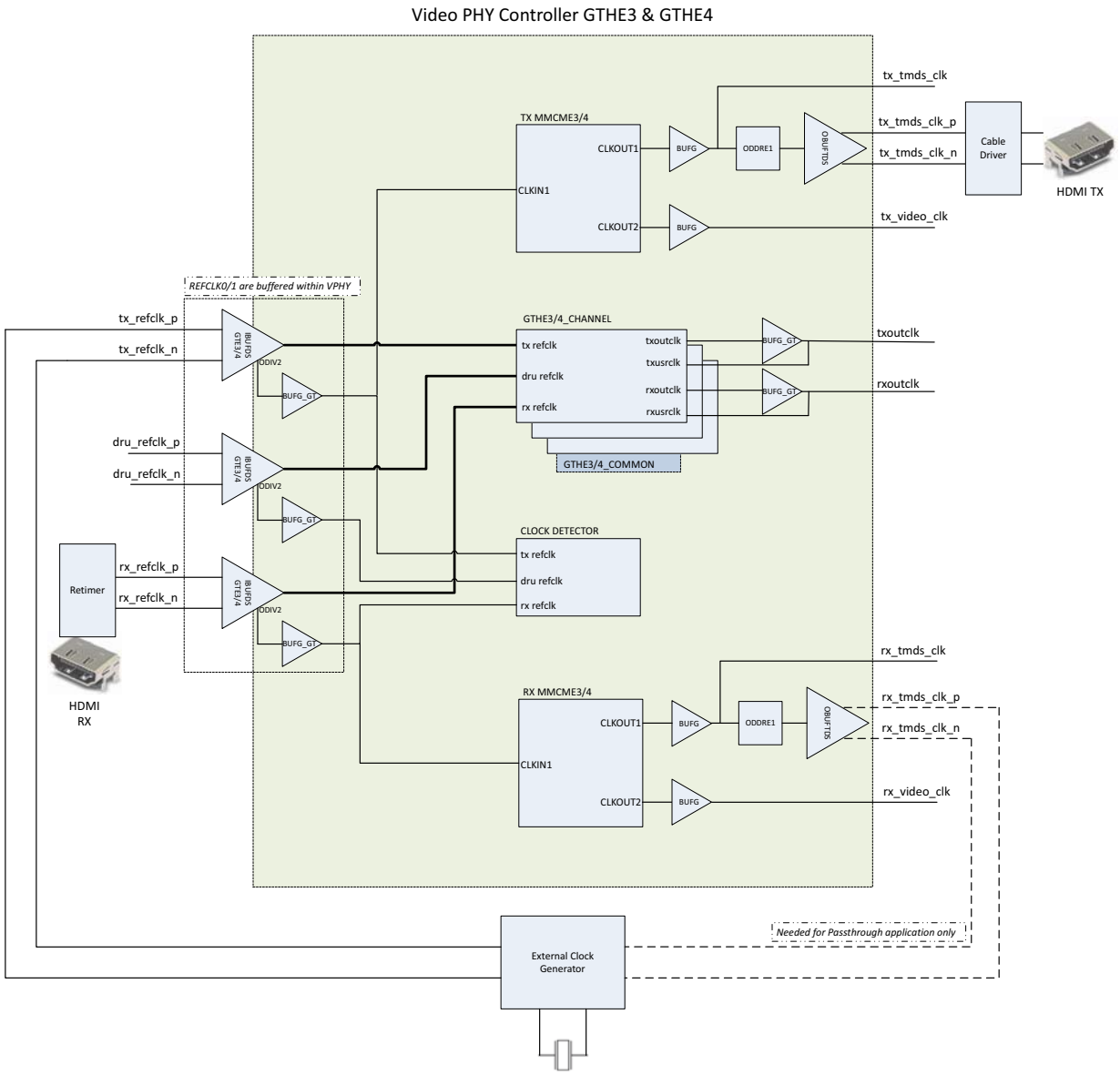

Figure 3-7: **GTHE3 and GTHE4**

Connecting UltraScale/UltraScale+ GT NORTH Reference Clock

Video PHY Controller opens the north and south reference clocks ports differently, depending on which PLL they are associated with. Additional ports with suffix 00 and 01 are opened when a reference clock is used with QPLL0/1.

When a reference clock is used with CPLL, the Video PHY Controller core opens two input ports, a GT channel or CPLL clock and a fabric clock (suffix odiv2). For example, NORTHREFCLK1 is used with CPLL, Video PHY Controller core opens the following input ports:

- gtnorthrefclk1_in: This port is the GT Channel or CPLL clock and must be connected to the IBUF_OUT port of the IBUFDSGTE GT input clock buffer. See Figure 3-8.
- gtnorthrefclk1_odiv2_in: This port is the fabric clock and must be connected to the BUFG_GT_O port of the BUFG_GT buffer which is being driven by the IBUF_DS_ODIV2 output of IBUFDSGTE GT input clock buffer. See Figure 3-8.

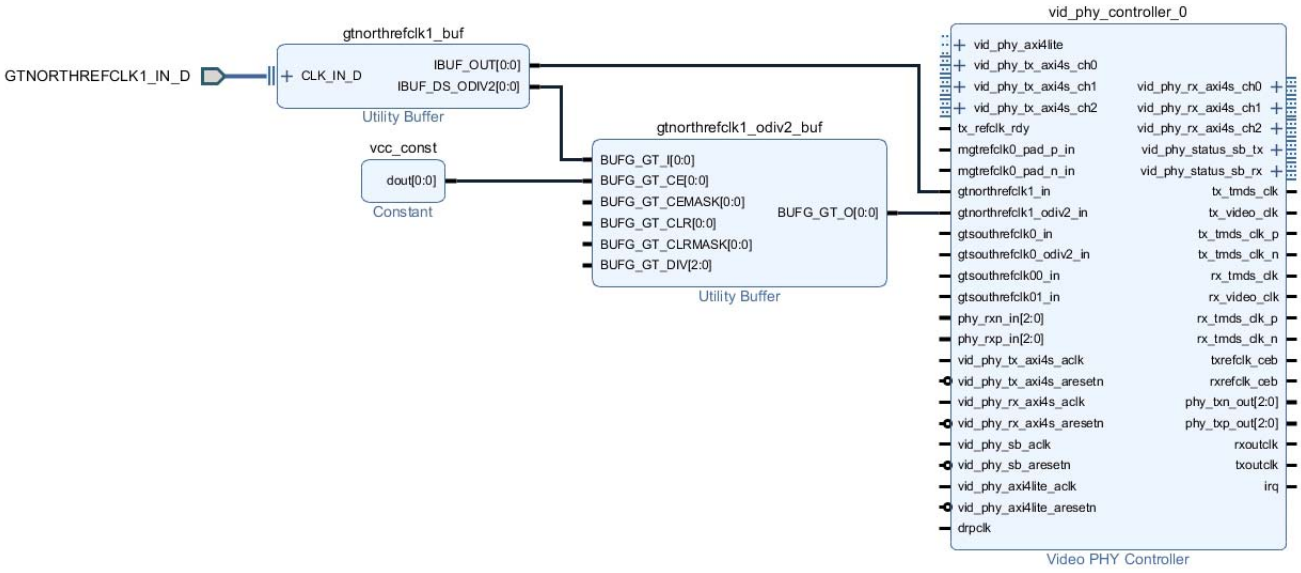

Figure 3-8: **North Reference Clock**

Notes:

1. BUFG_GT_CE port of the BUFG_GT buffer must be driven HIGH.
2. IBUF_DS_ODIV2 port of the IBUFDSGTE is by default configured to output divide by 1.

Connecting UltraScale/UltraScale+ GT South Reference Clock

When a reference clock is used with QPLL0/1, the Video PHY Controller core opens four input ports, a GT channel clock, two common clocks and a fabric clock (suffix odiv2). For example, SOUTHREFCLK0 is used with QPLL0/1, Video PHY Controller core opens the following input ports:

- gtsouthrefclk0: This port is the GT Channel clock and must be connected to the IBUF_OUT port of the IBUFDSGTE GT input clock buffer. See Figure 3-9.
- gtsouthrefclk00_in: This port is the QPLL0 clock and must be connected to the IBUF_OUT port of the IBUFDSGTE GT input clock buffer. See Figure 3-9.
- gtsouthrefclk01_in: This port is the QPLL1 clock and must be connected to the IBUF_OUT port of the IBUFDSGTE GT input clock buffer. See Figure 3-9.
- gtsouthrefclk0_odiv2_in: This port is the fabric clock and must be connected to the BUFG_GT_O port of the BUFG_GT buffer which is being driven by the IBUF_DS_ODIV2 output of IBUFDSGTE GT input clock buffer. See Figure 3-9.

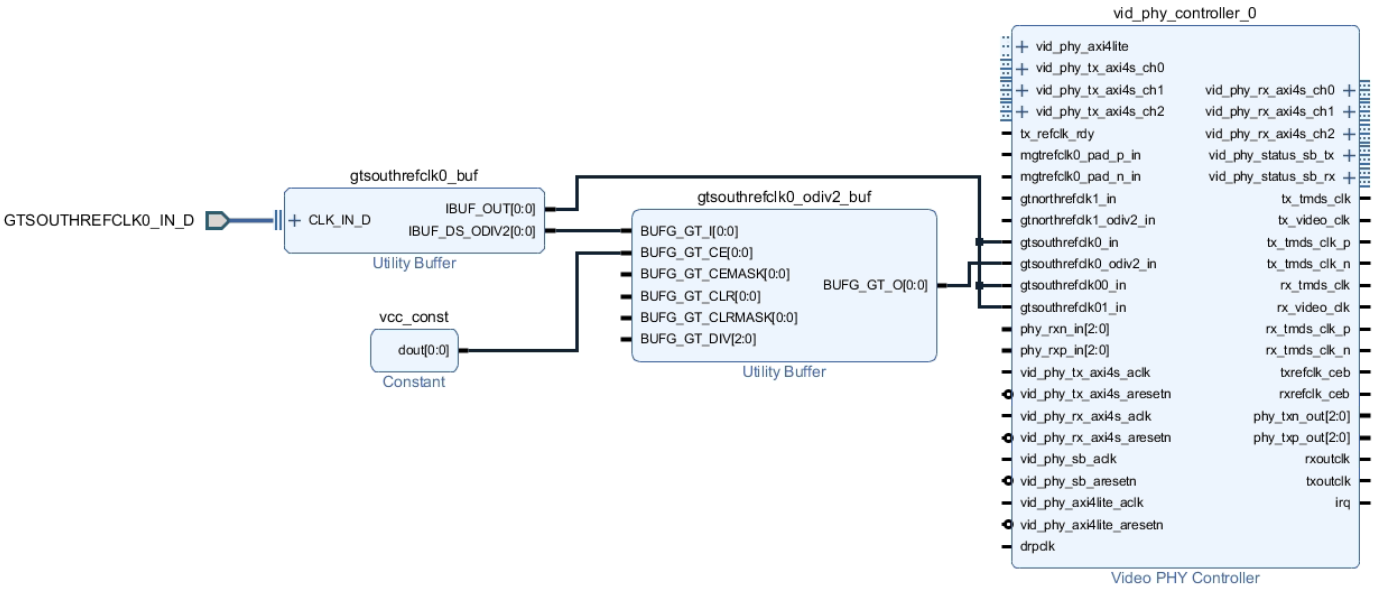

Figure 3-9: **South Reference Clock**

Notes:

1. BUFG_GT_CE port of the BUFG_GT buffer must be driven HIGH.
2. IBUF_DS_ODIV2 port of the IBUFDSGTE is by default configured to output divide by 1.

Resets

This core uses a GT Wizard to generate reset FSM. The reset for the Wizard FSM is provided as a software bit defined in the Register Map — TX Initialization (Address `0x001C`) and RX Initialization (Address `0x0024`).

Interrupts

This core issues an active High IRQ and must be used with an interrupt controller that is Level-High sensitive. Using sensitivity modes other than what was specified may result to incorrect software behavior.

Program and Interrupt Flow

Video PHY DisplayPort TX Flow

Figure 3-10 shows the Video PHY Controller core in the DisplayPort TX program flow.

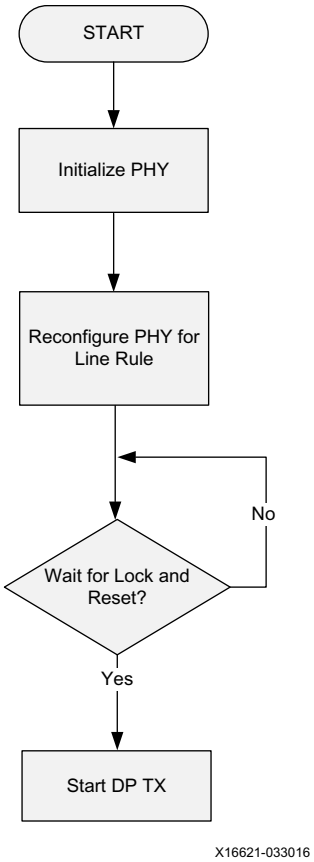

Figure 3-10: **Video PHY DisplayPort TX Program Flow**

Video PHY DisplayPort RX Flow

Figure 3-11 shows the Video PHY Controller core in the DisplayPort RX program flow.

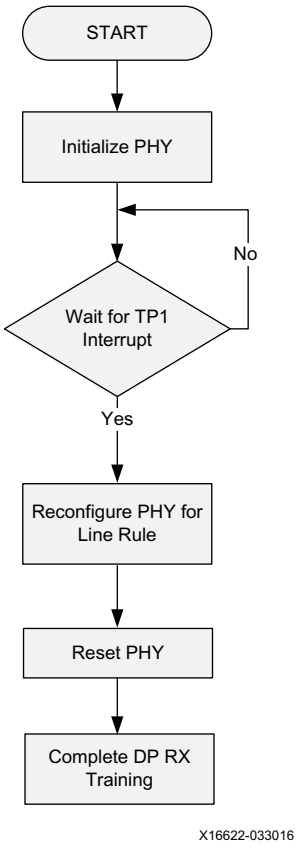

Figure 3-11: **Video PHY DisplayPort RX Program Flow**

HDMI Program and Interrupt Flow

The Video PHY Controller core driver manages the dynamic reconfiguration of the multi-gigabit transceiver and digital clock manager modules to allow seamless transmission and reception of HDMI video to and from the FPGA physical interface.

The main program flow is shown in Figure 3-12 and Figure 3-13. At execution, the software application initializes the Video PHY IP and registers the callback functions in the provided hooks. After the initialization, all API calls are interrupt triggered starting from either TX or RX reference clock change.

Note: The Video PHY Controller driver does not carry the video format, resolution, or color space information. Such information are handled by the HDMI TX and RX MAC. Refer to *HDMI 1.4/2.0 Transmitter Subsystem LogiCORE IP Product Guide* [Ref 17] and *HDMI 1.4/2.0 Receiver Subsystem LogiCORE IP Product Guide* [Ref 18] for more information.

Video PHY HDMI TX Flow

A change in TX reference clock signifies a video format change which triggers a series of interrupts until the GT TX attains the TX Alignment Done status. The TX frequency change is based on the toggling (deassertion then assertion) of the `tx_refclk_rdy` port. Refer to Video PHY HDMI Reference Clocks Requirements in Chapter 4 for details about `tx_refclk_rdy` port implementation.

There are several API callback hooks that the Video PHY Controller core execute throughout the HDMI TX operation. If necessary, these callbacks are available for inserting or adding more function calls on top of what is in the software application.

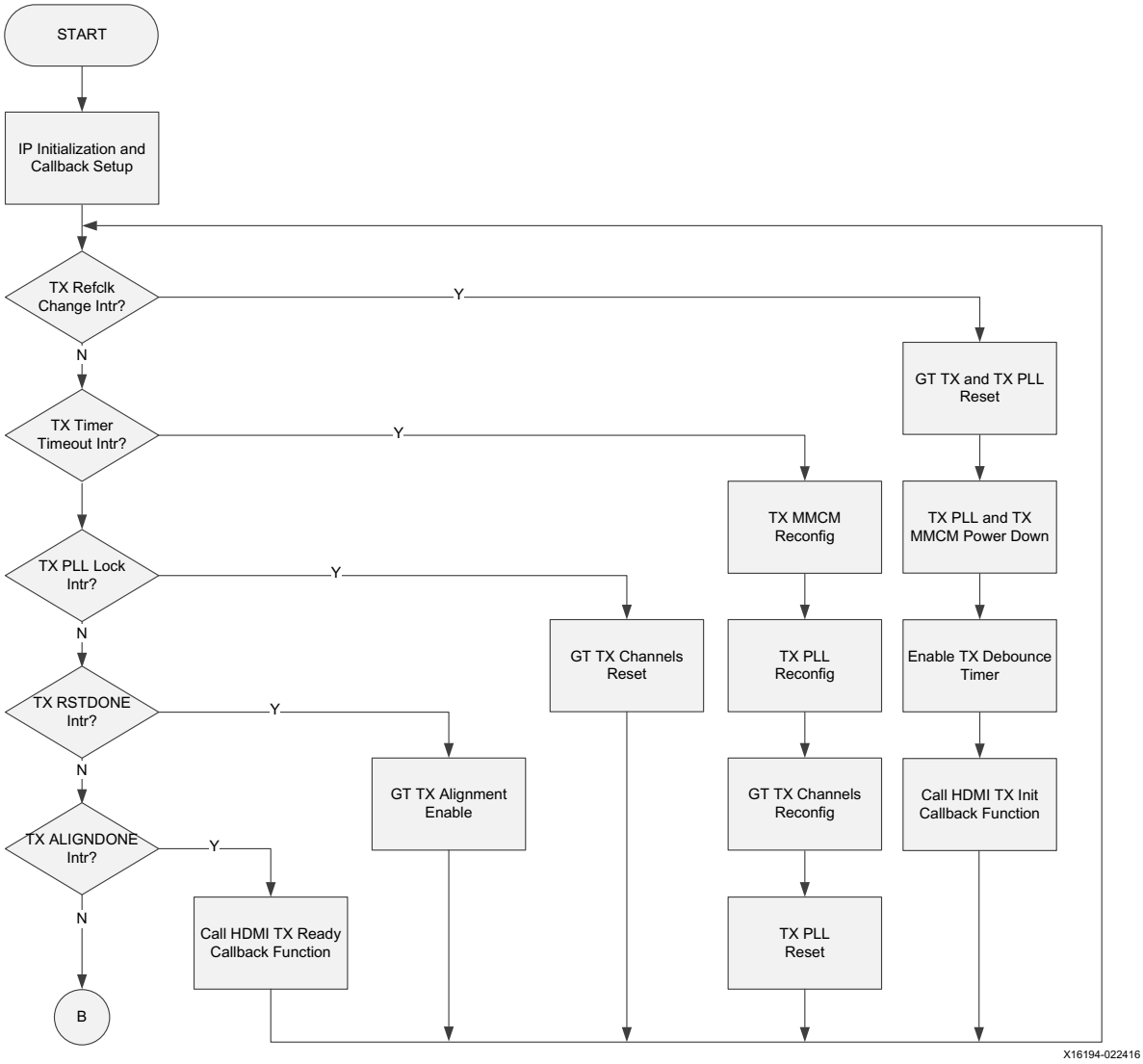

Figure 3-12: **Video PHY HDMI TX Program Flow**

Video PHY Controller Core Driver TX Callbacks

The Video PHY Controller core driver TX callbacks are:

- **Video PHY Controller HDMI TX Init Callback** – This callback is named XVPHY_HDMI_HANDLER_TXINIT in the Video PHY Controller core driver. It is executed or called every time a change in the HDMI TX Transition Minimized Differential Signaling (TMDS) clock frequency occurs. This normally triggers a reset to the HDMI TX Subsystem IP.
- **Video PHY Controller HDMI TX Ready Callback** – This callback is named XVPHY_HDMI_HANDLER_TXREADY in the Video PHY Controller core driver. It is executed or called every time the Video PHY Controller core driver completes the initialization routine necessary for the TX video format change.

Video PHY HDMI RX Flow

A change in RX reference clock signifies a video format change which triggers a series of interrupts until the GT RX attains the RX Reset Done status.

There are several API callback hooks that the Video PHY Controller core execute throughout the HDMI RX operation. If necessary, these callbacks are available for inserting or adding more function calls on top of what is in the software application.

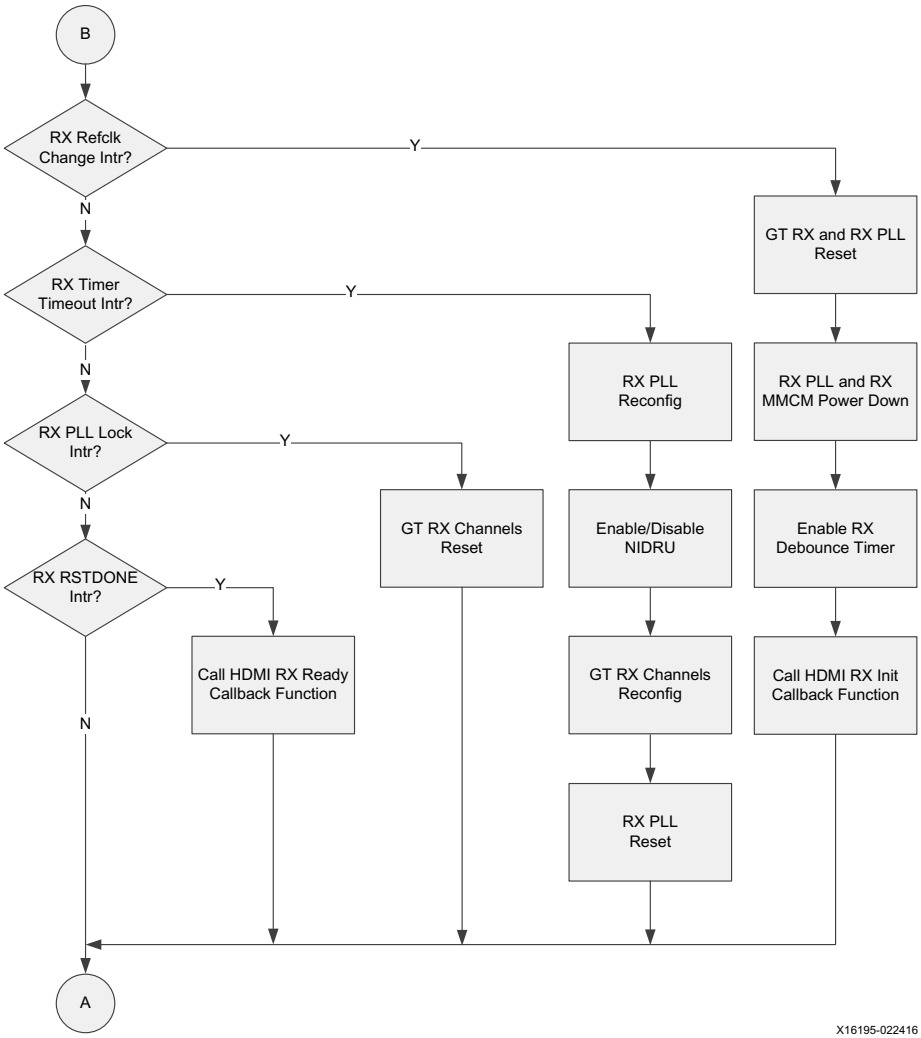

Figure 3-13: **Video PHY HDMI RX Program Flow**

Video PHY Controller Core Driver RX Callbacks

The Video PHY Controller core driver RX callbacks are:

- **Video PHY Controller HDMI RX Init Callback** – This callback is named XVPHY_HDMI_HANDLER_RXINIT in the Video PHY Controller core driver. It is executed or called every time a change in the HDMI RX TMDS clock frequency occurs.
- **Video PHY Controller HDMI RX Ready Callback** – This callback is named XVPHY_HDMI_HANDLER_RXREADY in the Video PHY Controller core driver. It is executed or called every time the Video PHY Controller core driver completes the initialization routine necessary for the RX video format change. The hook normally updates the clock and line rate parameters of the HDMI RX Subsystem IP.

Video PHY Controller HDMI Implementation

UltraScale GTHE3 and UltraScale+ GTHE4 Video PHY Controller HDMI Implementation

The GTHE3/4 transceiver in the UltraScale™/UltraScale+™ FPGAs has three types of PLLs, the QPLL0, QPLL1, and the CPLL. The QPLL0 and QPLL1 are shared by all four transceivers in the Quad. Each transceiver has its own CPLL. The Video PHY Controller core uses all of the PLL types to support transmitter and receiver operations simultaneously. The Video PHY Controller core allows you to choose whether the QPLL0/1 or the CPLL is used by the transmitter. The receiver should use the other PLL that is not used by TX.

Using the CPLL for the HDMI receiver includes certain restrictions. The TX does not have these restrictions because the GT driver uses oversampling techniques to work around the limitations of the CPLL. The HDMI RX limitations for the CPLL are described in this section.

The CPLL voltage controlled oscillator (VCO) must run in the range of 2.0 GHz to 6.25 GHz. The VCO frequency is dependent upon the TMDS clock frequency. The CPLL can apply a limited set of multipliers to the TMDS clock frequency. The GT driver measures the TMDS clock frequency and attempts to find a valid multiplier that results in a VCO frequency that is within the allowed range.

Because the largest multiplier that can be applied by the CPLL is 20, the minimum TMDS clock frequency that can be supported by the CPLL is 100 MHz. Video formats that have a TMDS clock frequency of less than 100 MHz are not supported by the CPLL.

When the GT driver detects that the TMDS clock frequency is less than 100 MHz, it enables the NI-DRU to receive these low bit rates that are less than 1 Gb/s. The NI-DRU runs at 2.5 Gb/s, which enables it to recover line rates that cannot be supported by the CPLL.

For TMDS clock frequencies greater than 100 MHz, a multiplier of 10X or 20X is applied to keep the VCO frequency in the proper range as shown in Table 3-1.

Table 3-1: **UltraScale GTH CPLL Usage**

TMDS Clock Frequency (MHz)	CPLL Refclk Divider	CPLL Multiplier	VCO Frequency	Notes
<100	N/A	N/A	N/A	TX: Oversampling RX: NI-DRU is used
100 to 312.5	1	20	2.0 to 6.25 GHz	CDR is used
312.5 to 340	1	10	3.125 to 3.4 GHz	CDR is used

IMPORTANT: *Using the CPLL, the HDMI RX can receive the most valid video format, using the NI-DRU or the native CDR, up to a maximum line rate of 6 Gb/s.*

A limitation of the CPLL is that it cannot be used to receive or transmit 2160p24/25/30 at 10 BPC because the TMDS reference clock for these formats is below 100 MHz and the line rate is above 2.5 Gb/s. Oversampling techniques cannot be used because of the limitation on the available CPLL divider values.

Table 3-2: **CPLL Support of RGB and YCbCr 4:4:4 Video Formats**

Resolution (Hz)	Bits Per Pixel			
	24	30	36	48
480i60	DRU	DRU	DRU	DRU
576i50	DRU	DRU	DRU	√
1080i50	DRU	DRU	√	√
1080i60	DRU	DRU	√	√
480p60	DRU	DRU	DRU	DRU
576p50	DRU	DRU	DRU	DRU
720p50	DRU	DRU	√	√
720p60	DRU	DRU	√	√
1080p24	DRU	DRU	√	√
1080p25	DRU	DRU	√	√
1080p30	DRU	DRU	√	√
1080p50	√	√	√	√
1080p60	√	√	√	√
2160p24	√	(1)	√	√
2160p25	√	(1)	√	√
2160p30	√	(1)	√	√
2160p60	√	(2)	(2)	(2)
VGA 60	DRU	DRU	DRU (4)	DRU (4)
SVGA 60	DRU	DRU	DRU	√
XGA 60	DRU	DRU	DRU	√
SXGA 60	√	√	√	√
WXGA 60	DRU	DRU	√	√
WXGA + 60	DRU	√	√	√
UXGA 60	√	√	√	(3)
WUXGA 60	√	√	√	√

Table 3-2: **CPLL Support of RGB and YCbCr 4:4:4 Video Formats *(Cont'd)***

Resolution (Hz)	Bits Per Pixel			
	24	30	36	48
WSXGA 60	√	√	√	√

Notes:

1. These formats are not supported in the CPLL because it falls in between the support range of the NI-DRU and the CPLL. Although x3 oversampling can be used to overcome the CPLL limitation for transmission, there is no Digital Clock Manager (DCM) divider settings available to generate the required TX Link, TX Video and TX TMDS clocks (see PG235) from the oversampled reference clock thus TX cannot support these formats. See H*DMI 1.4/2.0 Transmitter Subsystem Product Guide* (PG235) [Ref 17] for more information.
2. This format is not supported because it exceeds the maximum line rate of HDMI 2.0.
3. This format is supported for transmit, but is not currently supported by the receiver.
4. VGA 12 and 16 BPC at 4 PPC are not supported by TX due to DCM limitations.

When using the two Quad PLL (QPLL) types for the HDMI transmitter and receiver, line rate restrictions are introduced due to the VCO range and limited set of multipliers of the QPLL. The Video PHY Controller core driver dynamically switches between QPLL0 and QPLL1 to overcome these restrictions. For the transmitter, the Video PHY Controller core driver uses oversampling and the dynamic QPLL switching to work around the QPLL limitations.

The VCO of the QPLL0 must run in the frequency range of 9.8 GHz to 16.3 GHz. The QPLL0 can apply multipliers of 20, 40, 80, or 160 to the TMDS clock.

IMPORTANT: *The limited VCO range and the available multipliers of the QPLL0 cause gaps in the range of line rates that can be supported for the HDMI.*

Table 3-3 shows how the TMDS clock frequency interacts with the QPLL0 and the frequency ranges that can be supported.

Table 3-3: **UltraScale GTH QPLL0 Usage**

TMDS Clock Frequency (MHz)	QPLL0 Multiplier	Notes
<61.25	N/A	TX: Oversampling RX: NI-DRU is used
61.25 to 101.875	160	Supported
101.875 to 122.5	–	TMDS clock range cannot be supported
122.5 to 203.75	80	Supported
203.75 to 245	–	TMDS clock range cannot be supported
245 to 407	40	Supported
407 to 490	–	TMDS clock range cannot be supported

If the QPLL0 is used as the clock source, video formats with a TMDS clock of 101.875 MHz to 122.5 MHz, 203.75 MHz to 245 MHz, and frequencies higher than 407 MHz cannot be supported since a multiplier cannot be used to meet the valid VCO range.

The VCO of the QPLL1 must run in the frequency range of 8.0 GHz to 13.0 GHz. The QPLL1 can apply multipliers of 20, 40, 80, or 160 to the TMDS clock.

IMPORTANT: *The limited VCO range and the available multipliers of the QPLL1 cause gaps in the range of line rates that can be supported for the HDMI.*

Table 3-4 shows how the TMDS clock frequency interacts with the QPLL1 and the frequency ranges that can be supported.

If the QPLL1 is used as the clock source, video formats with a TMDS clock of 81.25 MHz to 100 MHz, 162.5 MHz to 200 MHz, and 325 MHz to 400 MHz cannot be received because a multiplier cannot be used to meet the valid VCO range.

Table 3-4: **UltraScale GTH QPLL1 Usage for HDMI RX**

TMDS Clock Frequency (MHz)	QPLL1 Multiplier	Notes
<50.0	N/A	TX: Oversampling RX: NI-DRU is used
50.0 to 81.25	160	Supported
81.25 to 100	–	TMDS clock range cannot be supported
100 to 162.5	80	Supported
162.5 to 200	–	TMDS clock range cannot be supported
200 to 325	40	Supported
325 to 400	–	TMDS clock range cannot be supported
400 to 650	20	Supported

Table 3-5: **QPLL Support of RGB and YCbCr 4:4:4 Video Formats**

Resolution (Hz)	Bits Per Pixel			
	24	30	36	48
480i60	DRU	DRU	DRU	DRU
576i50	DRU	√	√	√
1080i50	√	√	√	√
1080i60	√	√	√	√
480p60	DRU	DRU	DRU	DRU
576p50	DRU	DRU	DRU	DRU
720p50	√	√	√	√
720p60	√	√	√	√
1080p24	√	√	√	√
1080p25	√	√	√	√
1080p30	√	√	√	√

Table 3-5: **QPLL Support of RGB and YCbCr 4:4:4 Video Formats *(Cont'd)***

Resolution (Hz)	Bits Per Pixel			
	24	30	36	48
1080p50	√	√	√	√
1080p60	√	√	√	√
2160p24	√	√	√	√
2160p25	√	√	√	√
2160p30	√	√	√	√
2160p60	√	(1)	(1)	(1)
VGA 60	√	DRU	DRU	DRU
SVGA 60	√	DRU	DRU	√
XGA 60	√	√	√	√
SXGA 60	√	√	√	√
WXGA 60	√	√	√	√
WXGA + 60	√	√	√	√
UXGA 60	√	√	√	(2)
WUXGA 60	√	√	√	√
WSXGA 60	√	√	√	√

Notes:

1. This format is not supported because it exceeds the maximum line rate of HDMI 2.0.
2. This format is supported for transmit, but is not currently supported by the receiver.

7 Series GTXE2 Video PHY Controller Core HDMI Implementation

The GTX transceiver in 7 series FPGAs has two types of PLLs, the QPLL and the CPLL. The QPLL is shared by all four transceivers in the Quad. Each transceiver has its own CPLL. The Video PHY IP allows you to choose whether the QPLL or the CPLL is used by the transmitter. The receiver should use the other PLL that is not used by TX.

The GTX transceivers can either use the Quad PLL (QPLL) or the Channel PLL (CPLL) as the clock source for the RX and the TX. The RX and the TX can use the same PLL or different PLLs. If the same PLL is used, for example both use the CPLL, then they are "bonded" and must always run at exactly the same line rate.

The QPLL and the CPLL both have certain limitations. The QPLL has "holes" certain line rates that it cannot support due to the limited frequency range of its VCO. However, the QPLL can support lower TMDS clock frequencies than the CPLL.

The CPLL voltage controlled oscillator (VCO) must run in the range of 1.6 GHz to 3.3 GHz. The VCO frequency is dependent upon the TMDS clock frequency. The CPLL can apply a limited set of multipliers to the TMDS clock frequency. The GT driver measures the TMDS

clock frequency and attempts to find a valid multiplier that results in a VCO frequency that is within the allowed range.

Because the largest multiplier that can be applied by the CPLL is 20, the minimum TMDS clock frequency that can be supported by the CPLL is 80 MHz. Video formats that have a TMDS clock frequency of less than 80 MHz are not supported by the CPLL.

When the GT driver detects that the TMDS clock frequency is less than 80 MHz, it enables the NI-DRU to receive these low bit rates that are less than 0.8 Gb/s. The NI-DRU runs at 2.0 Gb/s, which enables it to recover line rates that cannot be supported by the CPLL.

For TMDS clock frequencies greater than 80 MHz, a multiplier of 10X or 20X is applied to keep the VCO frequency in the proper range as shown in Table 3-6.

Table 3-6: **7 Series GTX CPLL Usage**

TMDS Clock Frequency (MHz)	CPLL Refclk Divider	CPLL Multiplier	VCO Frequency	Notes
<80	N/A	N/A	N/A	TX: Oversampling RX: NI-DRU is used
80 to 165	1	20	1.6 to 3.3 GHz	CDR is used
165 to 330	1	10	1.65 to 3.3 GHz	CDR is used

IMPORTANT: *Using the CPLL, the HDMI RX can receive the most valid video format, using the NI-DRU or the native CDR, up to a maximum line rate of 6 Gb/s.*

The CPLL can support all video formats. It has no "holes." However, the CPLL does require the use of the NI-DRU to cover any format that has a TMDS clock below 80 MHz. Table 3-7 shows the standard formats that are supported by the CPLL, indicating which require the DRU.

Table 3-7: **CPLL Support of RGB and YCbCr 4:4:4 Video Formats**

Resolution (Hz)	Bits Per Pixel			
	24	30	36	48
480i60	DRU	DRU	DRU	DRU
576i50	DRU	DRU	DRU	DRU
1080i50	DRU	√	√	√
1080i60	DRU	√	√	√
480p60	DRU	DRU	DRU	DRU
576p50	DRU	DRU	DRU	DRU
720p50	DRU	√	√	√
720p60	DRU	√	√	√
1080p24	DRU	√	√	√
1080p25	DRU	√	√	√

Table 3-7: **CPLL Support of RGB and YCbCr 4:4:4 Video Formats *(Cont'd)***

Resolution (Hz)	Bits Per Pixel			
	24	30	36	48
1080p30	DRU	√	√	√
1080p50	√	√	√	√
1080p60	√	√	√	√
2160p24	√	√	√	√
2160p25	√	√	√	√
2160p30	√	√	√	√
2160p60	√	(1)	(1)	(1)
vgap60	DRU(3)	DRU	DRU	DRU (3)
svgap60	DRU	DRU	DRU	√
xgap60	DRU	√	√	√
sxgap60	√	√	√	√
wxgap60	DRU	√	√	√
wxga+p60	√	√	√	√
uxgap60	√	√	√	(2)
wuxgap60	√	√	√	√
wsxgap60	√	√	√	√

Notes:

1. This format is not supported because it exceeds the maximum line rate of HDMI 2.0.
2. This format is supported for transmit, but is not currently supported by the receiver.
3. VGA 8 and 16 BPC at 4 PPC are not supported by TX due to Digital Clock Manager limitations.

When the QPLL is used as the clock source for either the RX or TX, there are combinations of resolutions, color depth, and color space that cannot be supported because the QPLL cannot generate the necessary clock frequencies to support those video formats. HDMI uses the lower band of QPLL. The VCO of the QPLL must run in the frequency range of 5.93 GHz to 8.0 GHz. The QPLL can apply multipliers of 20, 40, or 80 to the TMDS clock.

Table 3-8 shows how the TMDS clock frequency interacts with the QPLL and the frequency ranges that can be supported.

Table 3-8: **7 Series GTX QPLL Usage**

TMDS Clock Frequency (MHz)	QPLL Multiplier	Notes
<74.125	N/A	TX: Oversampling (1) RX: NI-DRU is used
74.125 to 100	80	Supported
100 to 148.25	–	TMDS clock range cannot be supported
148.25 to 200	40	Supported

Table 3-8: **7 Series GTX QPLL Usage *(Cont'd)***

TMDS Clock Frequency (MHz)	QPLL Multiplier	Notes
200 to 296.5	–	TMDS clock range cannot be supported
296.5 to 400	20	Supported

Notes:

1. There are certain resolutions (for example, 480p60 12 BPC) that cannot be worked around by x3 or x5 oversampling because the oversampled reference clocks also falls into QPLL hole.

Note that the QPLL has two operating band. For HDMI, the QPLL is always used in lower band because the upper band is not available in all speed grades.

Table 3-9 shows which of the standard RGB and YCbCr 4:4:4 video formats are supported when using the QPLL. Formats shown with a check mark are supported by the QPLL. Formats with the "–" are not supported because they fall into QPLL holes. Formats that require the DRU to receive them are noted in the table.

Table 3-9: **QPLL Support of RGB and YCbCr 4:4:4 Video Formats**

Resolution (Hz)	Bits Per Pixel			
	24	30	36	48
480i60	DRU	DRU	DRU	DRU
576i50	DRU	DRU	DRU	DRU
1080i50	√	√	–	√
1080i60	√	√	–	√
480p60	DRU	DRU	DRU	DRU
576p50	DRU	DRU	DRU	DRU
720p50	√	√	–	√
720p60	√	√	–	√
1080p24	√	√	–	√
1080p25	√	√	–	√
1080p30	√	√	–	√
1080p50	√	√	–	√
1080p60	√	√	–	√
2160p24	√	√	–	√
2160p25	√	√	–	√
2160p30	√	√	–	√
2160p60	√	(1)	(1)	(1)
vgap60	DRU	DRU	DRU (3)	DRU (3)
svgap60	DRU	DRU	DRU	DRU
xgap60	DRU	√	√	–

Table 3-9: **QPLL Support of RGB and YCbCr 4:4:4 Video Formats *(Cont'd)***

Resolution (Hz)	Bits Per Pixel			
	24	30	36	48
sxgap60	–	–	√	–
wxgap60	DRU	√	–	–
wxga+p60	√	–	–	√
uxgap60	√	–	–	(2)
wuxgap60	√	–	–	√
wsxgap60	–	√	√	–

Notes:

1. This format is not supported because it exceeds the maximum line rate of HDMI 2.0.
2. This format is supported for transmit, but is not currently supported by the receiver.
3. VGA 12 and 16 BPC at 4 PPC are not supported by TX due to Digital Clock Manager limitations.

7 Series GTPE2 Video PHY Controller Core HDMI Implementation

The GTP transceiver in 7 series FPGAs has two types of PLLs, the PLL0 and PLL1 of similar characteristics. Both PLL types are shared by all four transceivers in the Quad. There is no dedicated PLL for each transceiver channel for GTPE2. The Video PHY IP allows you to choose whether the PLL0 or the PLL1 is used by the transmitter. The receiver should use the other PLL that is not used by TX.

The PLL0/1 voltage controlled oscillator (VCO) must run in the range of 1.6 GHz to 3.3 GHz. The VCO frequency is dependent upon the TMDS clock frequency. The PLL0/1 can apply a limited set of multipliers to the TMDS clock frequency. The GT driver measures the TMDS clock frequency and attempts to find a valid multiplier that results in a VCO frequency that is within the allowed range.

Because the largest multiplier that can be applied by the PLL0/1 is 20, the minimum TMDS clock frequency that can be supported by the PLL0/1 is 80 MHz. Video formats that have a TMDS clock frequency of less than 80 MHz are not supported by the PLL0/1.

When the GT driver detects that the TMDS clock frequency is less than 80 MHz, it enables the NI-DRU to receive these low bit rates that are less than 0.8 Gb/s. The NI-DRU runs at 2.0 Gb/s, which enables it to recover line rates that cannot be supported by the PLL0/1. For

TMDS clock frequencies greater than 80 MHz, a multiplier of 10X or 20X is applied to keep the VCO frequency in the proper range as shown in Table 3-10.

Table 3-10: **7 Series GTP PLL0/1 Usage**

TMDS Clock Frequency (MHz)	PLL0/1 Refclk Divider	PLL0/1 Multiplier	VCO Frequency	Notes
<80	N/A	N/A	N/A	TX: Oversampling RX: NI-DRU is used
80 to 165	1	20	1.6 to 3.3 GHz	CDR is used
165 to 330	1	10	1.65 to 3.3 GHz	CDR is used

IMPORTANT: *Using the PLL0/1, the HDMI RX can receive the most valid video format, using the NI-DRU or the native CDR, up to a maximum line rate of 6 Gb/s. Parts with -1, -1L and -2LE (0.9V) speed grades are not supported by the HDMI Video PHY Controller because the maximum line rate for those devices is only 3.75 Gbps.*

The PLL0/1 can support all video formats. It has no "holes." However, the PLL0/1 does require the use of the NI-DRU to cover any format that has a TMDS clock below 80 MHz. Table 3-11 shows the standard formats that are supported by the PLL0/1, indicating which require the DRU.

Table 3-11: **PLL0/1 Support of RGB and YCbCr 4:4:4 Video Formats**

Resolution (Hz)	Bits Per Pixel			
	24	30	36	48
480i60	DRU	DRU	DRU	DRU
576i50	DRU	DRU	DRU	DRU
1080i50	DRU	√	√	√
1080i60	DRU	√	√	√
480p60	DRU	DRU	DRU	DRU
576p50	DRU	DRU	DRU	DRU
720p50	DRU	√	√	√
720p60	DRU	√	√	√
1080p24	√	√	√	√
1080p25	√	√	√	√
1080p30	√	√	√	√
1080p50	√	√	√	√
1080p60	√	√	√	√
2160p24	√	√	√	√
2160p25	√	√	√	√
2160p30	√	√	√	√

Table 3-11: **PLL0/1 Support of RGB and YCbCr 4:4:4 Video Formats *(Cont'd)***

Resolution (Hz)	Bits Per Pixel			
	24	30	36	48
2160p60	(4)	(1)	(1)	(1)
vgap60	DRU	DRU	DRU (3)	DRU (3)
svgap60	DRU	DRU	DRU	√
xgap60	DRU	√	√	√
sxgap60	√	√	√	√
wxgap60	DRU	√	√	√
wxga+p60	√	√	√	√
uxgap60	√	√	√	(2)
wuxgap60	√	√	√	√
wsxgap60	√	√	√	√

Notes:

1. This format is not supported because it exceeds the maximum line rate of HDMI 2.0.
2. This format is supported for transmit, but is not currently supported by the receiver.
3. VGA 12 and 16 BPC at 4 PPC are not supported by TX due to Digital Clock Manager limitations.
4. 2160p60 is only supported when Number of pixels per clock is 4.

Design Flow Steps

This chapter describes customizing and generating the core, constraining the core, and the simulation, synthesis and implementation steps that are specific to this IP core. More detailed information about the standard Vivado® design flows and the 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 10]
- *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 11]
- *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 12]
- *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 13]

Customizing and Generating the Core

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

If you are customizing and generating the core in the Vivado IP integrator, see the *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994) [Ref 10] 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 IP for use in your design by specifying values for the various parameters associated with the IP core using the following steps:

1. Select the IP from the Vivado IP catalog.
2. Double-click the selected IP or select the **Customize IP** command from the toolbar or right-click menu.

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 11] and the *Vivado Design Suite User Guide: Getting Started* (UG910) [Ref 12].

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

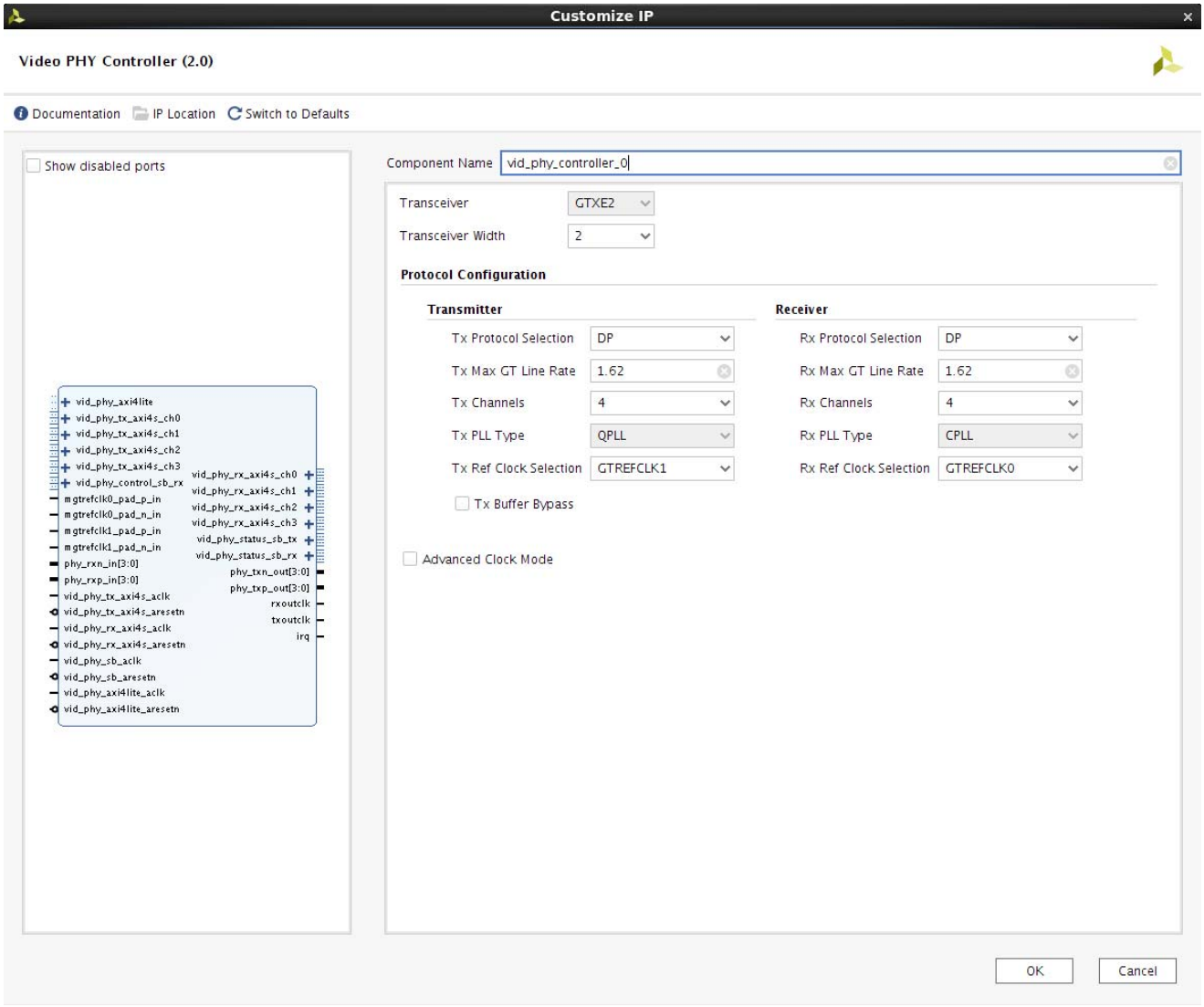

Figure 4-1: **Vivado IDE – DisplayPort Protocol with GTPE2, GTXE2, and GTHE2 Transceivers**

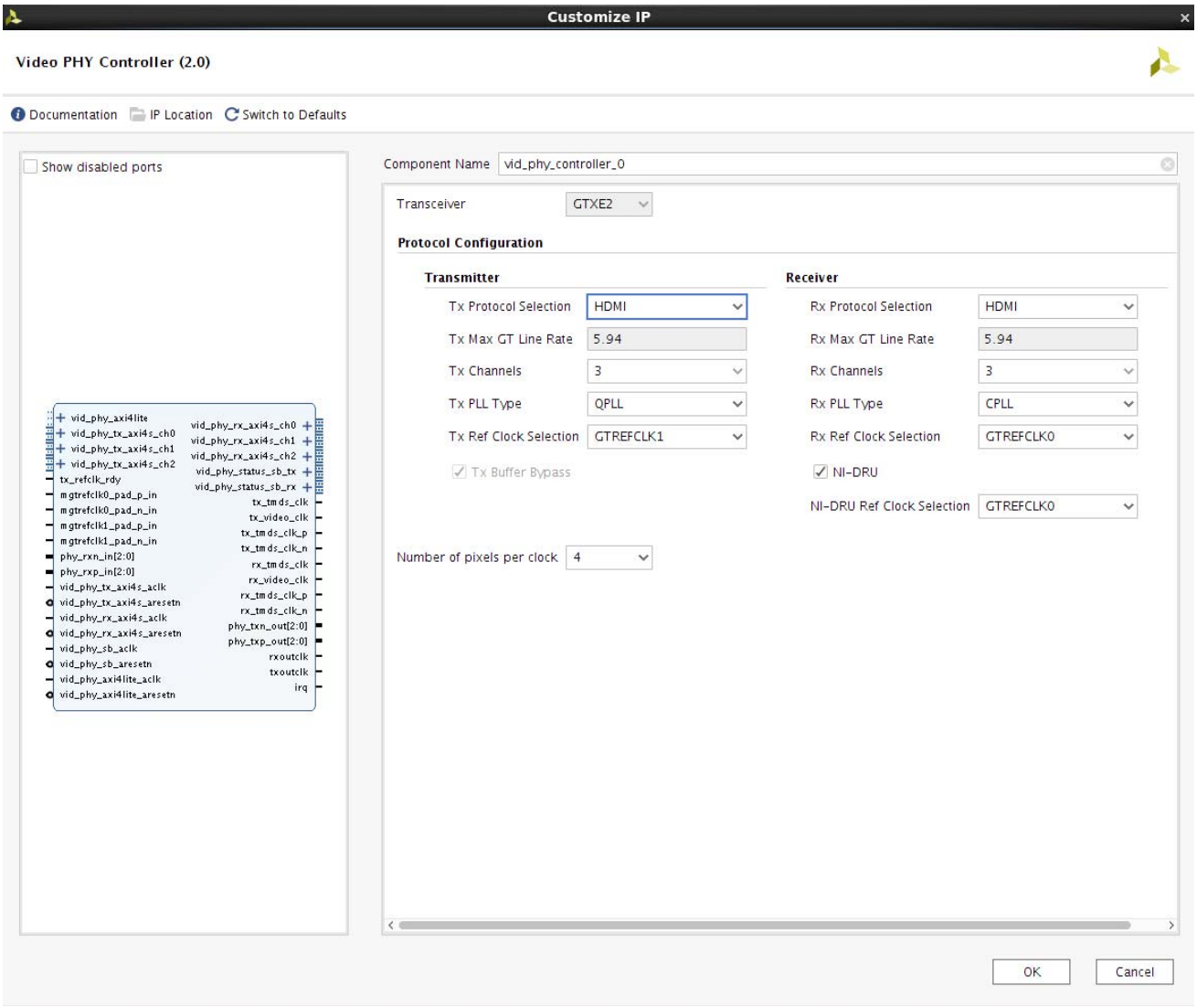

Figure 4-2: **Vivado IDE – HDMI Protocol with GTPE2 and GTXE2 Transceivers**

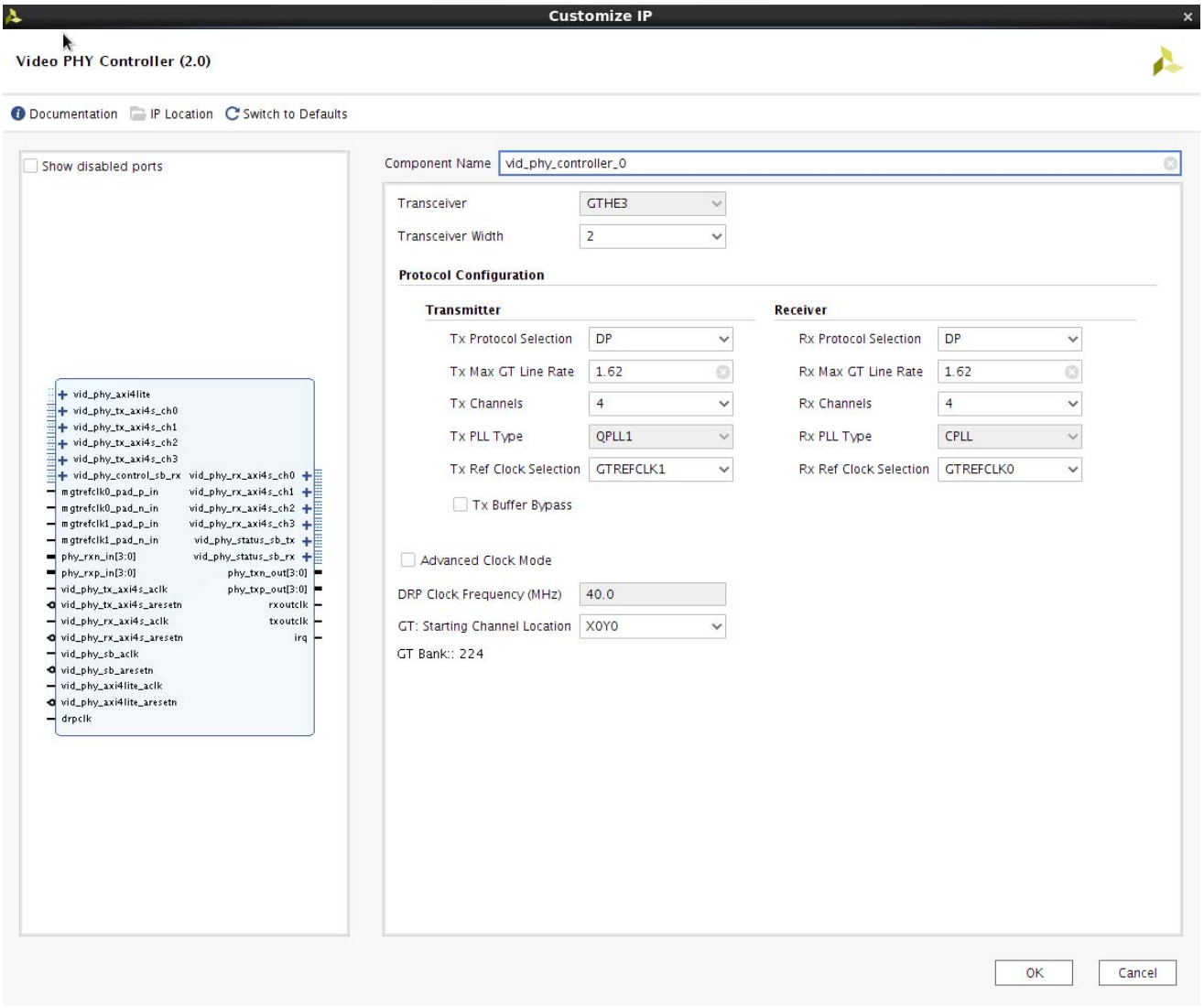

Figure 4-3: **Vivado IDE – DisplayPort Protocol with GTHE3 Transceiver**

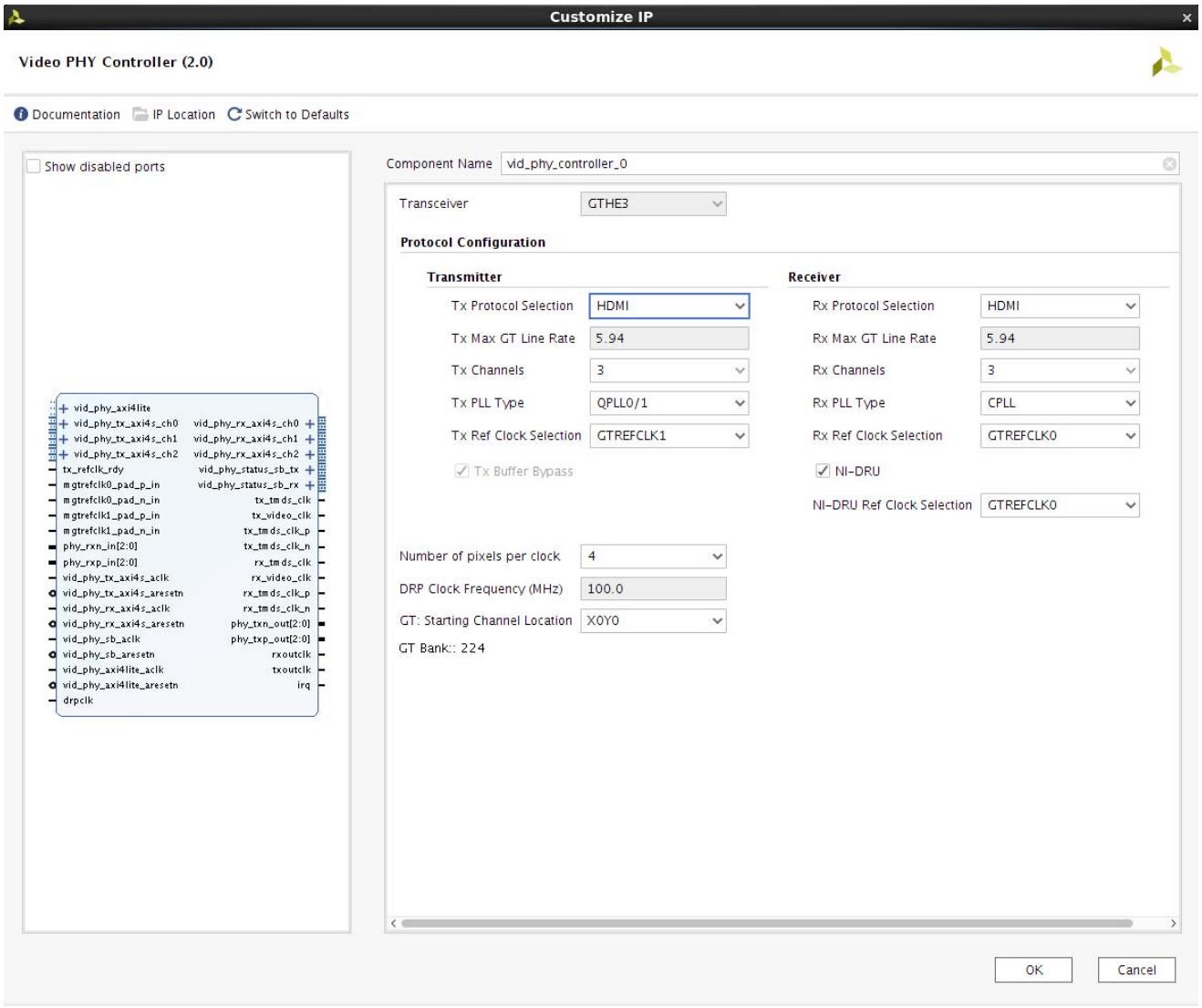

Figure 4-4: **Vivado IDE – HDMI Protocol with GTHE3 and GTHE4 Transceivers**

The Vivado IDE displays a representation of the IP symbol on the left side, and the parameter assignments on the right side which are described as follows:

Component Name: The component name is used as the base name of the output files generated for the module. Names must begin with a letter and must be composed from characters: a to z, 0 to 9 and "_". The name `vid_phy_controller` cannot be used as a component name.

Transceiver: Specifies the types of transceiver that is used in this core.

Transceiver Width: Specifies the width of the transceiver that is used in this core (Option is displayed when DisplayPort protocol is selected. See Figure 4-1 or Figure 4-3).

Tx/Rx Protocol Selection: Specifies the protocol that is supported under this core. Two protocols are currently available: DisplayPort and HDMI® (High-Definition Multimedia

Interface). Mixed protocols is not supported for both transmitter and receiver (that is, transceiver protocol is HDMI and receiver protocol is DisplayPort, and vice-versa).

Note: When **Tx/Rx Protocol Selection** is set to **None**, some of the options such as **PLL type and Ref Clock Selection** are still open for changes and is vary per protocol of opposite direction. These options can be ignored when **Tx/Rx Protocol Selection** is set to **None**. Considered the following scenarios as examples:

- GTHE3 with Tx is **DisplayPort** and Rx is **None**.
  - The only configurable option in Rx is **Rx Ref Clock Selection** but this has no impact because both `refclk` ports are open for Tx usage regardless the setting in Rx Ref Clock Selection.
- GTHE3 with Tx is **HDMI** and Rx is **None**.
  - The only configurable options are **Rx PLL Type** and **Rx Ref Clock Selection**.
  - There is automatic checking on **PLL Type** which disallow the setting of the same PLL type for Tx and Rx.
  - For **Rx Ref Clock Selection**, the setting has no impact on the `refclk` port.

Tx/Rx Max GT Line Rate: Specifies the maximum line rate for the transceiver. For HDMI protocol, this option is fixed to 5.94 Gbps.

Tx/Rx Channel: Specifies the number of transceiver channels to be generated in this core. For DisplayPort protocol, this option is allowed to have one, two, or four channels. For HDMI protocol, this option is fixed to three channels only.

Tx/Rx Ref Clock Selection: Specifies the reference clock that corresponds to the transceiver.

Tx Buffer Bypass: When checked, the Tx buffer is excluded in the core.

Ni-DRU: When checked, the NI-DRU is included in the core. (Option is displayed when HDMI protocol is selected for receiver. See Figure 4-2 or Figure 4-4.)

Ni-DRU Ref Clock Selection: Specifies the reference clock that corresponds to the NI-DRU. (Option is displayed when HDMI protocol is selected for receiver. See Figure 4-2 or Figure 4-4.)

Number of pixels per clock: Specifies the number of pixels for video clock generation. (Option is displayed when HDMI protocol is selected. See Figure 4-2 or Figure 4-4.)

Advanced Clock Mode: When checked, the core exposes most of the available clock ports (for example, `gtnorth/southrefclk0/1_ports` are revealed in GTHE3/4 transceiver type). When HDMI protocol is selected, this option is disabled.

DRP Clock Frequency (MHz): Specifies the frequency that needs to be driven at the DRP clock. This option is displayed when **UltraScale Transceiver** is selected.

GT: Starting Channel Location Specifies the starting channel location that aligns with the Quad boundary. This option is displayed when **UltraScale Transceiver** is selected.

GT Bank <num>: Indicates the transceiver bank location. (This option Indicator is displayed when **UltraScale Transceiver** is selected.

User Parameters

Table 4-1 shows the relationship between the 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**

Vivado IDE Parameter/Value	User Parameter/Value	Default Value
Tx/Rx Protocol Selection	C_Tx/Rx_Protocol	DisplayPort
DisplayPort		
HDMI		
None		
Tx/Rx Max GT Line Rate	Rx_Max_GT_Line_Rate	1.62
Tx/Rx Channels	C_Tx/Rx_No_Of_Channels	4
Tx PLL Type	C_TX_PLL_SELECTION	3
CPLL: 0		
QPLL0: 1		
QPLL1: 2		
QPLL: 3		
PLL0: 4		
PLL1: 5		
QPLL0/1: 6		
Rx PLL Type	C_RX_PLL_SELECTION	0
(similar to TX PLL Type)		
Tx Ref Clock Selection	C_TX_REFCLK_SEL	1
GTREFCLK0: 0		
GTREFCLK1: 1		
GTNORTHREFCLK0: 2		
GTNORTHREFCLK1: 3		
GTSOUTHREFCLK0: 4		
GTSOUTHREFCLK1: 5		
GTEASTREFCLK0: 6		
GTEASTREFCLK1: 7		
GTWESTREFCLK0: 8		
GTWESTREFCLK1: 9		
Rx Ref Clock Selection	C_RX_REFCLK_SEL	0

Table 4-1: **Vivado IDE Parameter to User Parameter Relationship *(Cont'd)***

Vivado IDE Parameter/Value	User Parameter/Value	Default Value
(similar to Tx Ref Clock Selection)		
Tx Buffer Bypass	Tx_Buffer_Bypass	false
NI-DRU	C_NIDRU	true
NI-DRU Ref Clock Selection	C_NIDRU_REFCLK_SEL	0
(similar to Tx Ref Clock Selection)		
Advanced Clock Mode	Adv_Clk_Mode	false
Number of pixels per clock Value Selection	C_INPUT_PIXELS_PER_CLOCK	2
1		
2		
4		
DRP Clock Frequency (MHz)	DRPCLK_FREQ	40.0
Transceiver Width Value Selection	Transceiver_Width	2
2		
4		

Output Generation

For details, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 11].

Constraining the Core

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

Out-of-Context Constraints

When an out-of-context (OOC) design flow such as OOC synthesis or hierarchical design is used, the PHY also uses a special OOC XDC file customized for that instance. The OOC XDC file provides default period constraints on clock ports that would otherwise be constrained by the System XDC file.

Required Constraints

For GTXE2, you must add GT clocking and location constraints. For GTHE3, location constraints are added according to the VIvado Bank customization. You should add clocking and location constraints.

Device, Package, and Speed Grade Selections

The core constraints generated for a given instance reflect the selections made during IP customization for the target device. If you wish to use a different device, package, or speed grade, use the Vivado IDE to select the desired part and re-customize the core rather than modifying an XDC file.

Clock Frequencies

Video PHY HDMI Reference Clocks Requirements

The Video PHY HDMI application requires a system clock and a maximum of three GT reference clock inputs:

- System Clock
- HDMI TX from an external clock generator
- HDMI RX in CDR mode (normal operation)
- HDMI RX NI-DRU mode

The System Clock should drive the `vid_phy_sb_aclk`, `vid_phy_axi4lite_aclk`, and `drpclk` (for UltraScale+™/UltraScale™ devices only) ports should be connected to a 100 MHz clock.

The HDMI TX and RX reference clock (Transition Minimized Differential Signaling (TMDS) clocks) input frequency varies according to the input video and both are maxed at 297 MHz.

The NI-DRU reference clock frequency is fixed and is dependent on the transceiver type as follows:

- GTXE2: 125 MHz
- GTHE3 and GTHE4: 156.25 MHz
- GTPE2: 100 MHz

Note: Although theoretically a vast range of REFCLK frequencies can be used with NI-DRU, only the indicated frequencies have been tested and characterized per transceiver type. The NI-DRU settings such as gain were optimized and validated using the indicated frequencies.

Figure 4-5 illustrates the full reference clock requirement connections.

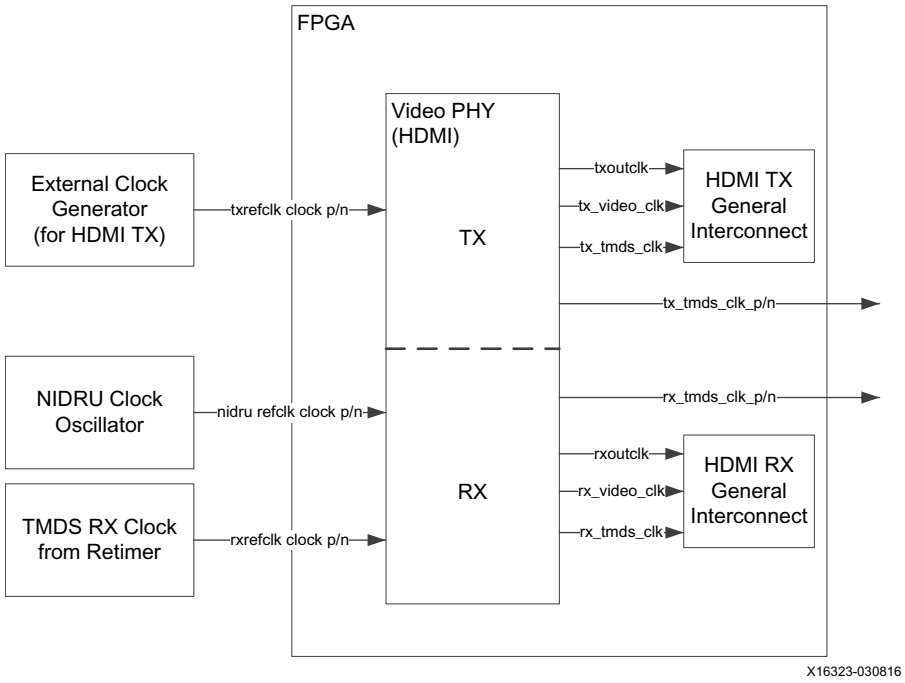

Figure 4-5: **Video PHY HDMI Reference Clock Connections**

The HDMI TX reference clock comes from an external programmable clock generator capable of generating a range of frequencies from minimum PLL reference clock (see Table 4-2) to maximum TMDS clock for supported video formats. For example, the Video PHY TX is using GTXE2 QPLL and supporting up to 4Kp60 at two pixels per clock. This means the programmable clock generator must be able to generate frequencies from 74.125 MHz to 297 MHz. For resolutions requiring lower TMDS clock than the minimum PLL reference clock, Video PHY uses the oversampling technique (see the following section for details).

IMPORTANT: *The txrefclk port is accompanied by tx_refclk_rdy port to indicate a lock condition. The tx_refclk_rdy port has two requirements.*

The `tx_refclk_rdy` port requirements:

- Connected to the external clock generator's lock pin by default or can be toggled through GPIO. It must be toggled (de-asserted then asserted) for every video format change.
- It can ONLY be asserted when the clock at `txrefclk_p/n` port is stable.

IMPORTANT: *Failing to meet these two requirements causes instability to the system. This means that the tx_refclk_rdy port cannot be permanently tied High to the logic as it violates the first rule.*

TX REFCLK frequency detection is sensitive only to the behavior of the `tx_refclk_rdy` port, which triggers the Clock Detector to issue the TX frequency change event. This is because users program the external clock generator for the desired clock frequency in the TX operation, which means that the VPHY TX should get the requested frequency from the clock generator. Because the assumption is that VPHY gets the correct clock, it only requires the LOCK event to trigger the TX reconfiguration which is represented by the assertion of `tx_refclk_rdy`.

Note: This mechanism applies only for TX. RX is sensitive to the frequency change because there is no user control on the incoming RX TMDS clock.

Video PHY HDMI TX Oversampled Reference Clock Requirements

In normal cases, the GT reference clock requirement[1] is equal to the TMDS clock requirement of a given HDMI resolution. The Video PHY HDMI TX application enters the oversampling mode when the reference clock required by video resolution to be transmitted is below the HDMI PLL minimum frequency. The Video PHY driver increases the reference clock by a factor of x3 or x5 until the minimum frequency for PLL is met. Table 4-2 shows the minimum reference clock per transceiver and PLL types.

For example, GTHE3 CPLL needs to transmit 480p 60Hz at eight bits per component. This video format requires a TMDS clock of 27 MHz which is below the GTHE3 CPLL 100 MHz minimum clock. The Video PHY driver searches for the oversampling factor that satisfies this condition, which is x5. The new GT reference clock requirement is 135 MHz.

Table 4-2: **HDMI Transceiver to PLL Type Minimum Reference Clock**

Transceiver Type	PLL Type	HDMI Min Reference Clock (MHz)
GTHE3/GTHE4	QPLL0	61.25
	QPLL1	50
	CPLL	100
GTXE2	QPLL	74.125
	CPLL	80
GTPE2	PLL0/PLL1	80

Video PHY HDMI TX Clock Requirement Example

The frequency range of the external programmable clock generator must be selected based on the transceiver type and the PLL type for TX. Table 4-3 shows the frequency range needed from the clock generator if the Video PHY is used to support all the video formats in Table 3-2, Table 3-5, Table 3-7, Table 3-9, and Table 3-11.

Table 4-3: **External Clock Generator Typical Frequency Range**

Configuration	TX PLL	Reference Clock Range (MHz)
GTHE3/GTHE4	CPLL	100 to 297
	QPLL0	61.25 to 297
GTXE2	CPLL	80 to 297
	QPLL	74.125 to 297
GTPE2	PLL0/1	80 to 297

1. The GT reference clock requirement value can be accessed through the HdmiTxRefClkHz variable in the Video PHY data structure declared in the application (for example, in reference design: Vphy.HdmiTxRefClkHz). The HdmiTxRefClkHz value is valid and can be accessed any time after TX Alignment Done Interrupt occurs (see the Video PHY HDMI TX Flow in Chapter 3). This value is ideally used in programming the external clock generator frequencies for GT TX operation.

Table 4-4 shows the external clock generator frequency range if the Video PHY TX is used to support video formats of SMPTE-SDI: SD-SDI, HD-SDI, and 3G-SDI which in HDMI have equivalent TMDS clocks 27, 74.25, or 74.25/1.001 MHz and 148.5 or 148.5/1.001 MHz, respectively. SD-SDI reference clock is below the minimum threshold of all PLL types thus oversampling mode must be used to support it. HD-SDI reference clock is below the minimum threshold of GTHE3/GTHE4, GTXE2 CPLL, and GTPE2 PLL0/1 thus oversampling mode must be used to support it for corresponding GT and PLL types.

Table 4-4: **External Clock Generator Frequency Range for SMPTE-SDI**

Transceiver Type	TX PLL	Reference Clock Range (MHz)	Remarks
GTHE3/GTHE4	CPLL	135 to 222.75	SD-SDI uses x5 oversampling HD-SDI uses x3 oversampling
	QPLL0	74.25/1.001 to 148.5	SD-SDI uses x3 oversampling
GTXE2	CPLL	81 to 222.75	SD-SDI uses x3 oversampling HD-SDI uses x3 oversampling
	QPLL	74.25/1.001 to 148.5	SD-SDI uses x3 oversampling
GTPE2	PLL0/1	81 to 222.75	SD-SDI uses x3 oversampling HD-SDI uses x3 oversampling

Video PHY HDMI Generated Clocks

The Video PHY Controller IP generates the TX TMDS, link and video clocks that are required by HDMI 1.4/2.0 Transmitter Subsystem. See the *Clocking* section of *HDMI 1.4/2.0 Transceiver Subsystem LogiCORE IP Product Guide* (PG235) [Ref 17] for more information.

The Video PHY Controller IP generates the RX link and video clocks that are required by HDMI 1.4/2.0 Receiver Subsystem. See the *Clocking* section of *HDMI 1.4/2.0 Receiver Subsystem LogiCORE IP Product Guide* (PG236) [Ref 18] for more information.

Clock Management

This section is not applicable for this IP core.

Clock Placement

For 7 series devices, you are expected to create package pin constraints for each instantiated transceiver differential reference clock buffer primitive as well as each instantiated differential recovered clock output buffer primitive, if utilized. The constraints reflect the transceiver primitive site locations.

Banking

Video PHY Controller does not support multiple GT bank/quad in one IP instance. Multiple VPHY instances is needed for applications requiring more than one active GT bank/quad.

Note: The majority of APIs in VPHY driver include the "QuadId" argument. This must be permanently set to "0", because VPHY only supports one GT bank/quad per instance.

Transceiver Placement

For 7 series, you must create an XDC file with location constraints for each enabled transceiver channel primitive. The constraints reflect the transceiver primitive site locations. For UltraScale and UltraScale+ devices, Vivado IDE provides customization for Transceiver Placement.

I/O Standard and Placement

DisplayPort

This section contains details about I/O constraints.

AUX Channel

The *VESA DisplayPort Standard* [Ref 1] describes the AUX channel as a bidirectional LVDS signal. For 7 series designs, the core uses IOBUFDS (bi-directional buffer) as the default with the LVDS standard. You should design the board as recommended by the VESA DisplayPort Protocol Standard. For reference, see the example design XDC file.

For Kintex®-7 and Artix®-7 devices supporting HR IO banks, use the following constraints:

For Source:

```
set_property IOSTANDARD LVDS_25   [get_ports aux_tx_io_p]
set_property IOSTANDARD LVDS_25   [get_ports aux_tx_io_n]
```

For Sink:

```
set_property IOSTANDARD LVDS_25   [get_ports aux_rx_io_p]
set_property IOSTANDARD LVDS_25   [get_ports aux_rx_io_n]
```

For Kintex-7 and Virtex®-7 devices supporting HP IO banks, use the following constraints:

For Source:

```
set_property IOSTANDARD LVDS [get_ports aux_tx_io_p]
set_property IOSTANDARD LVDS [get_ports aux_tx_io_n]
```

For Sink:

```
set_property IOSTANDARD LVDS   [get_ports aux_rx_io_p]
set_property IOSTANDARD LVDS   [get_ports aux_rx_io_n]
```

HPD

The HPD signal can operate in either a 3.3V or 2.5V I/O bank. By definition in the standard, it is a 3.3V signal.

For Kintex-7 and Artix-7 devices supporting HR IO banks, use the following constraints:

```
set_property IOSTANDARD LVCMOS25 [get_ports hpd];
```

For Virtex-7 devices supporting HP IO banks, use the following constraints:

```
set_property IOSTANDARD LVCMOS18 [get_ports hpd];
```

Board design and connectivity should follow *DisplayPort Standard* recommendations with proper level shifting.

High-Speed I/O

The four high-speed lanes operate in the LVDS (LVDS25) IO standard. Board design and connectivity should follow DisplayPort standard recommendations.

HDMI

DDC

The DDC I2C signals are implemented as bidirectional signals that use IOBUF in the FPGA. These signals can operate in either a 2.5V or 1.8V I/O bank. By definition in the standard, these are 3.3V signals.

For Kintex-7 and Artix-7 devices supporting HR IO banks, use the following constraints:

IO Standard:

```
set_property IOSTANDARD LVCMOS25 [get_ports tx_ddc_out_sda_io]
set_property IOSTANDARD LVCMOS25 [get_ports tx_ddc_out_scl_io]
set_property IOSTANDARD LVCMOS25 [get_ports rx_ddc_out_sda_io]
set_property IOSTANDARD LVCMOS25 [get_ports rx_ddc_out_scl_io]
```

Sample Pin Assignments:

```
set_property PACKAGE_PIN B17 [get_ports tx_ddc_out_sda_io]
set_property PACKAGE_PIN C17 [get_ports tx_ddc_out_scl_io]
set_property PACKAGE_PIN A27 [get_ports rx_ddc_out_sda_io]
set_property PACKAGE_PIN B27 [get_ports rx_ddc_out_scl_io]
```

For Virtex-7, UltraScale and UltraScale+ devices supporting HP IO banks, use the following constraints:

IO Standard:

```
set_property IOSTANDARD LVCMOS18 [get_ports tx_ddc_out_sda_io]
```

```
set_property IOSTANDARD LVCMOS18 [get_ports tx_ddc_out_scl_io]
set_property IOSTANDARD LVCMOS18 [get_ports rx_ddc_out_sda_io]
set_property IOSTANDARD LVCMOS18 [get_ports rx_ddc_out_scl_io]
```

Sample Pin Assignments:

```
set_property PACKAGE_PIN A20 [get_ports tx_ddc_out_sda_io]
set_property PACKAGE_PIN B20 [get_ports tx_ddc_out_scl_io]
set_property PACKAGE_PIN A9 [get_ports rx_ddc_out_sda_io]
set_property PACKAGE_PIN B9 [get_ports rx_ddc_out_scl_io]
```

Board design and connectivity should follow HDMI Standard recommendations with proper level shifting.

HPD

The HPD signal can operate in either a 2.5V or 1.8V I/O bank. By definition in the standard, it is a 5V signal.

For Kintex-7 and Artix-7 devices supporting HR IO banks, use the following constraints:

IO Standard:

```
set_property IOSTANDARD LVCMOS25 [get_ports TX_HPD_IN]
set_property IOSTANDARD LVCMOS25 [get_ports RX_HPD_OUT]
```

Sample Pin Assignments:

```
set_property PACKAGE_PIN F22 [get_ports TX_HPD_IN]
set_property PACKAGE_PIN D19 [get_ports RX_HPD_OUT]
```

For Virtex-7, UltraScale and UltraScale+ devices supporting HP IO banks, use the following constraints:

IO Standard:

```
set_property IOSTANDARD LVCMOS18 [get_ports TX_HPD_IN]
set_property IOSTANDARD LVCMOS18 [get_ports RX_HPD_OUT]
```

Sample Pin Assignments:

```
set_property PACKAGE_PIN A25 [get_ports TX_HPD_IN]
set_property PACKAGE_PIN A24 [get_ports RX_HPD_OUT]
```

Board design and connectivity should follow HDMI Standard recommendations with proper level shifting.

TMDS Clock

The TX TMDS Clock Output is implemented as LVDS (LVDS25) IO standard. It is important to note that VPHY currently does not support the usage of a GT TX channel as TMDS clock source.

The RX TMDS and NI_DRU Clock Inputs is implemented as a GT Reference Clock Input thus IO standard constraints are not required.

For Kintex-7 and Artix-7 devices, use the following constraints:

IO Standard:

```
TX TMDS: set_property IOSTANDARD LVDS_25 [get_ports HDMI_TX_CLK_P_OUT]
RX TMDS & NI-DRU: N/A
```

Sample Pin Assignments:

```
TX TMDS: set_property PACKAGE_PIN C19 [get_ports HDMI_TX_CLK_P_OUT]
RX TMDS: set_property PACKAGE_PIN C8 [get_ports HDMI_RX_CLK_P_IN]
NI-DRU:  set_property PACKAGE_PIN G8 [get_ports DRU_CLK_IN_clk_p]
```

For Virtex-7, UltraScale and UltraScale+ devices, use the following constraints:

IO Standard:

```
TX TMDS: set_property IOSTANDARD LVDS [get_ports HDMI_TX_CLK_P_OUT]
RX TMDS & NI-DRU: N/A
```

Sample Pin Assignments:

```
TX TMDS: set_property PACKAGE_PIN H21 [get_ports HDMI_TX_CLK_P_OUT]
RX TMDS: set_property PACKAGE_PIN C8 [get_ports HDMI_RX_CLK_P_IN]
NI-DRU:  set_property PACKAGE_PIN G8 [get_ports DRU_CLK_IN_clk_p]
```

Board design and connectivity should follow HDMI Standard recommendations with proper level shifting or TMDS driver usage.

High-Speed I/O

The three differential pairs of TX and RX high-speed lanes are implemented as GT TX and RX channels respectively thus IO standard constraints are not required. Board design and connectivity should follow HDMI standard recommendations.

For 7 series devices, actual pin assignments are required. Use the following constraints:

Sample Pin Assignments:

```
set_property PACKAGE_PIN E4 [get_ports {HDMI_RX_DAT_P_IN[0]}]
set_property PACKAGE_PIN D6 [get_ports {HDMI_RX_DAT_P_IN[1]}]
set_property PACKAGE_PIN B6 [get_ports {HDMI_RX_DAT_P_IN[2]}]
```

For UltraScale and UltraScale+ devices actual pin assignments are absorbed by the GT Wizard instance in the VPHY thus pin assignment constraints are not required:

Sample Pin Assignments:

```
N/A
```

Board Design Guidelines

HDMI

Transmitter

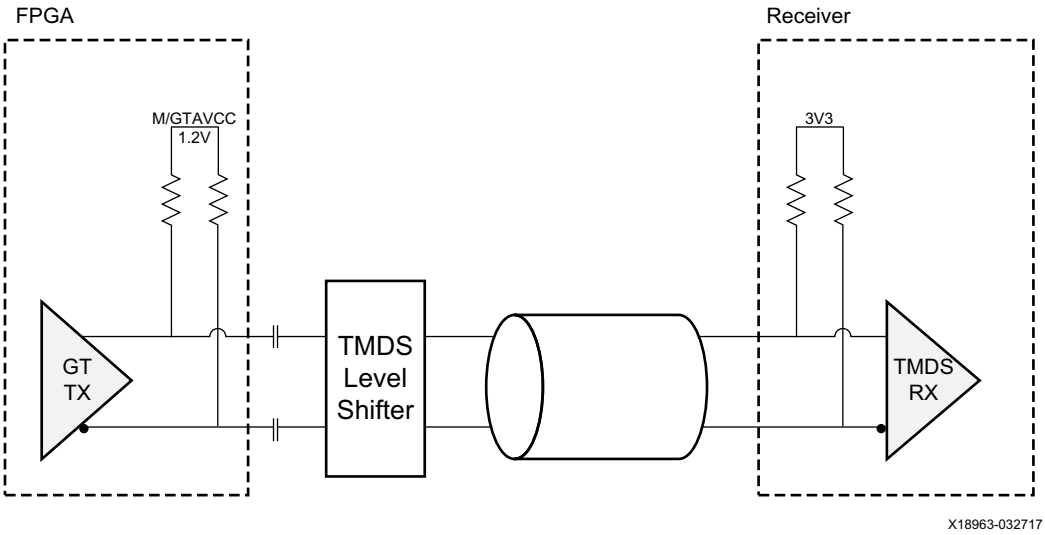

Figure 4-6: **Transmitter Board**

- GT Transmitter do not support TMDS Level Signaling and must be used with a cable driver to be compliant with TMDS specification.
- TMDS Level Shifting can be done using external level shifter ASSPs such as:
  - Texas Instruments SN65DP159
  - Parade Technology PS8409
- Board design simulation must be done to ensure proper operation.

Receiver

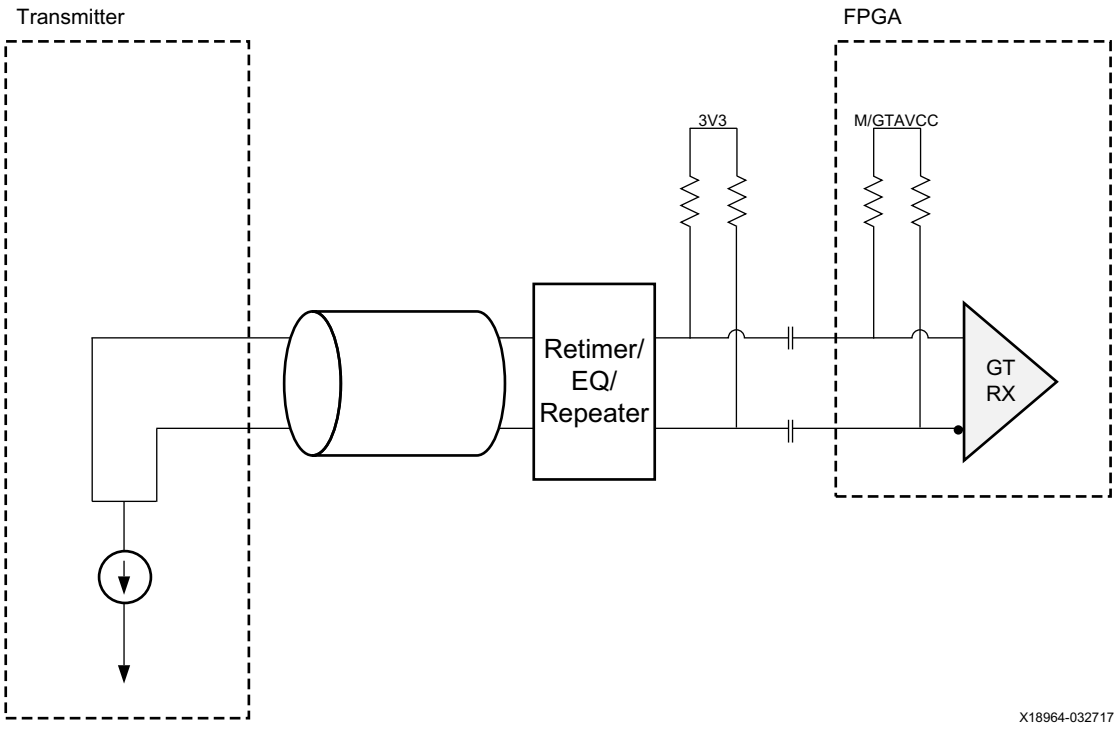

Figure 4-7: **Receiver Board**

- GT Receiver do not support TMDS Level Signaling and must be used with a retimer or equalizer to be compliant with TMDS specification.
- TMDS levels are emulated using external pull-up resistors (located close to the FPGA GTs).
- Use external TMDS Retimer/EQ chip recommended for HDMI 2.0 data rates such as:
  - Texas Instruments TMDS181.
  - Parade Technology PS8409.
- Board design simulation must be done to ensure proper operation.

Simulation

For comprehensive information about Vivado simulation components, as well as information about using supported third-party tools, see the *Vivado Design Suite User Guide: Logic Simulation* (UG900) [Ref 13].

IMPORTANT: *For cores targeting 7 series or Zynq-7000 devices, UNIFAST libraries are not supported. Xilinx IP is tested and qualified with UNISIM libraries only.*

Synthesis and Implementation

For details about synthesis and implementation, see the *Vivado Design Suite User Guide: Designing with IP* (UG896) [Ref 11].

Example Design

HDMI VPHY Example Design

Refer to *Application Software Development* section of *HDMI 1.4/2.0 Transmitter Subsystem Product Guide* (PG235) [Ref 17] and *HDMI 1.4/2.0 Receiver Subsystem Product Guide* (PG236) [Ref 18] for details on running the HDMI Example Design flow.

Note: HDMI Example Design is only available for KC705, KCU105, ZC706 and ZCU102 development boards.

Note: Ensure to update both IP and SW driver version when migrating designs to Vivado & SDK 2017.1. Failing to do so will cause compilation errors in SDK project.

DisplayPort VPHY Example Design

Refer to *DisplayPort Pass-Through Reference Design for SST and MST modes using Video PHY Controller* (XAPP1271) [Ref 19] and *DisplayPort Reference Design for SST and MST Modes* (XAPP1178) [Ref 20] for example.

Appendix A

Verification, Compliance, and Interoperability

This appendix provides details about how this IP core was tested for compliance with the protocol to which it was designed.

Simulation

The Video PHY Controller core uses the GT Wizard. The Video PHY Controller core is thoroughly simulated using an internal test bench for different configurations.

Hardware Testing

For 7 series devices, the KC 705 board is used. For UltraScale™ devices, the KCU 105 board is used.

Migrating and Upgrading

This appendix contains information about upgrading to a more recent version of the IP core.

This appendix does not apply to this core.

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 Licensing and Ordering Information in Chapter 1 for more details.*

Finding Help on Xilinx.com

To help in the design and debug process when using the core, 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.

Known Issues

Clock override critical warnings on TXOUTCLK or RXOUTCLK GT pins when compiling the DisplayPort Video PHY Controller are known issue and can be ignored. This is necessary in constraining the GT pins per target maximum line rate. Below is an example of the critical warning:

```
[Constraints 18-1056] Clock '<VPHY Path>/gtxe2_i/TXOUTCLK' completely overrides
clock '<VPHY Path>/gtxe2_i/TXOUTCLK'.
New: create_clock -period 3.704 [get_pins [list <VPHY Path>/gtxe2_i/TXOUTCLK <VPHY
Path>/gtxe2_i/TXOUTCLK <VPHY Path>/gtxe2_i/TXOUTCLK <VPHY Path>/gtxe2_i/TXOUTCLK]],
["<Project Path>/project_1.srcs/sources_1/bd/design_1/ip/
design_1_vid_phy_controller_0_0/vid_phy_controller_xdc.xdc": and 43]
Previous: create_clock -period 24.692 [get_pins <VPHY Path>/gtxe2_i/TXOUTCLK],
["<Project Path>/project_1.srcs/sources_1/bd/design_1/ip/
design_1_vid_phy_controller_0_0/ip_0/
design_1_vid_phy_controller_0_0_gtwrapper.xdc": and 83]
```

Documentation

This product guide is the main document associated with the core. 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 Video PHY Controller Core

AR: 57842

Technical Support

Xilinx provides technical support at 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.

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:

- Integrated logic analyzer (ILA) 2.0 (and later versions)
- Virtual Input Output (VIO) 2.0 (and later versions)

See the *Vivado Design Suite User Guide: Programming and Debugging* (UG908) [Ref 15].

Reference Boards

Various Xilinx development boards support the core. These boards can be used to prototype designs and establish that the core can communicate with the system.

- 7 series FPGA evaluation boards: KC705
- UltraScale™ device evaluation board: KCU105

Interface Debug

AXI4-Lite Interfaces

Read from a register that does not have all 0s as a default to verify that the interface is functional. Output `s_axi_arready` asserts when the read address is valid, and output `s_axi_rvalid` asserts when the read data/response is valid. If the interface is unresponsive, ensure that the following conditions are met:

- The `s_axi_aclk` and `aclk` inputs are connected and toggling.
- The interface is not being held in reset, and `s_axi_areset` is an active-Low reset.
- The interface is enabled, and `s_axi_aclken` is active-High (if used).
- The main core clocks are toggling and that the enables are also asserted.
- If the simulation has been run, verify in simulation and/or the Vivado Design Suite debug feature capture that the waveform is correct for accessing the AXI4-Lite interface.

AXI4-Stream Interfaces

If data is not being transmitted or received, check the following conditions:

- If transmit `<interface_name>_tready` is stuck Low following the `<interface_name>_tvalid` input being asserted, the core cannot send data.
- If the receive `<interface_name>_tvalid` is stuck Low, the core is not receiving data.
- Check that the `aclk` inputs are connected and toggling.
- Check that the AXI4-Stream waveforms are being followed.
- Check core configuration.

Application Software Development

For HDMI application software development, refer to *Application Software Development* section of *HDMI 1.4/2.0 Transmitter Subsystem Product Guide* (PG235) [Ref 17] and *HDMI 1.4/2.0 Receiver Subsystem Product Guide* (PG236) [Ref 18].

Additional Resources and Legal Notices

Xilinx Resources

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

Documentation Navigator and Design Hubs

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

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

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

- In the Xilinx Documentation Navigator, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

Note: For more information on Documentation Navigator, see the Documentation Navigator page on the Xilinx website.

References

These documents provide supplemental material useful with this product guide:

1. *VESA DisplayPort Standard v1.1a, January 11, 2008*
2. *Virtex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS893)
3. *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS892)
4. *Kintex-7 FPGAs Data Sheet: DC and AC Switching Characteristics* (DS182)
5. *Virtex-7 T and XT FPGAs Data Sheet: DC and AC Switching Characteristics* (DS183)
6. *Zynq UltraScale+ MPSoC Data Sheet: DC and AC Switching Characteristics* (DS925)
7. *Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics* (DS181)
8. *7 Series FPGAs GTX/GTH Transceivers User Guide* (UG476)
9. *UltraScale Architecture GTH Transceivers User Guide* (UG576)
10. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
11. *Vivado Design Suite User Guide: Designing with IP* (UG896)
12. *Vivado Design Suite User Guide: Getting Started* (UG910)
13. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
14. *ISE to Vivado Design Suite Migration Guide* (UG911)
15. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
16. *Vivado Design Suite User Guide: Implementation* (UG904)
17. *HDMI 1.4/2.0 Transmitter Subsystem LogiCORE IP Product Guide* (PG235)
18. *HDMI 1.4/2.0 Receiver Subsystem LogiCORE IP Product Guide* (PG236)
19. *DisplayPort Pass-Through Reference Design for SST and MST modes using Video PHY Controller* (XAPP1271)
20. *DisplayPort Reference Design for SST and MST Modes* (XAPP1178)

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
07/14/2017	2.0	• Updated Artix-7 support in IP Facts.
04/05/2017	2.0	• Added Clocking section. • Added Video PHY HDMI Generated Clocks section. • Updated Bank section. • Updated I/O Standard and Placement section.
06/08/2016	2.0	• Updated for GTHE4 support.
04/06/2016	2.0	• Updated Features in IP Facts. • Updated Unsupported Features in Overview chapter. • Updated NI-DRU and Performance sections in Product Specification chapter. • Updated User Clock Source section. • Updated PHY Controller Ports table. • Updated Register Map table. • Added DisplayPort Clocking in Clocking section. • Added Program and Interrupt Flow and Video PHY Controller HDMI Implementation sections in Designing with the Core chapter. • Updated Design Flow Steps chapter.
11/18/2015	2.0	Added HDMI® protocol support for 7 series devices (except Artix®-7) and Kintex® UltraScale™ devices.
09/30/2015	1.0	Early Access release.

Please Read: Important Legal Notices

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

applications, please refer to Xilinx’s Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

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