Video PHY Controller v2.2

LogiCORE IP Product Guide

Vivado Design Suite

PG230 (v2.2) June 30, 2021

Table of Contents

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. The interface between the video MAC and PHY layers are standardized to enable ease of use in accessing shared transceiver resources. The AXI4-Lite interface is provided to enable dynamic accesses of transceiver controls/status.

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

Features

- Protocol Support for DisplayPort and HDMI
- Protocol specific functions for HDMI
  - HDMI clock detector
  - Use of fourth GT channel as TX TMDS clock source
  - Non-integer data recovery unit (NI-DRU) support for lower line rates NI-DRU support is for the HDMI protocol only.
- 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
- Full transceiver dynamic reconfiguration port (DRP) accesses and transceiver functions
- Advanced clocking support
- AXI4-Lite support for register accesses

IP Facts

LogiCORE™ IP Facts Table	
Core Specifics	
Supported Device Family[1]	HDMI Video PHY Controller: UltraScale+™ families (GTHE4, GTYE4)[2] UltraScale™ families (GTHE3) Zynq®-7000 SoC (GTXE2) 7 series (GTXE2) Artix-7 (GTPE2)[7]
	DisplayPort Video PHY Controller: UltraScale+™ families (GTHE4, GTYE4)[2] UltraScale™ families (GTHE3) Zynq®-7000 SoC (GTXE2) Virtex®-7 (GTXE2) Kintex®-7 (GTXE2)[3]
Supported User Interfaces	AXI4-Stream, AXI4-Lite
Resources	Performance and Resource Use web page
Provided with Core	
Design Files	Verilog
Example Design	Provided with the HDMI and Display Port IP cores.[4]
Test Bench	Not Provided
Constraints File	Xilinx Design Constraints (XDC)
Simulation Model	Not Provided
Supported S/W Driver[5]	Standalone, Linux (HDMI Only)
Tested Design Flows[6]	
Design Entry	Vivado® Design Suite
Simulation	For supported simulators, see the Xilinx Design Tools: Release Notes Guide.
Synthesis	Vivado Synthesis
Support	
Release Notes and Known Issues	Master Answer Record: 57842
All Vivado IP Change Logs	Master Vivado IP Change Logs: 72775
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 Gb/s for GTHE4/GTYE4 Zynq® UltraScale+™ MPSoC and UltraScale+™ family -1LI (0.72 V) devices operated at 16-bit or 20-bit internal datapath.

   The maximum line rate for DisplayPort is 5.4 Gb/s for GTHE4/ GTYE4 Zynq® UltraScale+™ MPSoC, UltraScale+™ and GTHE3/ GTYE3 Zynq® UltraScale+™ MPSoC and UltraScale™ family -2LE (0.72 V), -1E,-1I, and -1LI (0.85 V) devices.
3. For Kintex-7: QPLL1/2 does not cover all DP line rate ranges. Using the CPLL for TX path is recommended.
4. See the HDMI and DisplayPort documentation references in References.
5. Standalone driver details can be found in <install_directory>/Vitis/<release>/data/embeddedsw/doc/xilinx_drivers_api_toc.htm.
6. For the supported versions of third-party tools, see the Xilinx Design Tools: Release Notes Guide.
7. GTPE2 -1, -1L, and -2LE (0.9V) parts are not supported by the Video PHY Controller. Artix -2 and -3 are limited to HDMI1.4 data rates due to fabric performance limitations.

Chapter 2

Overview

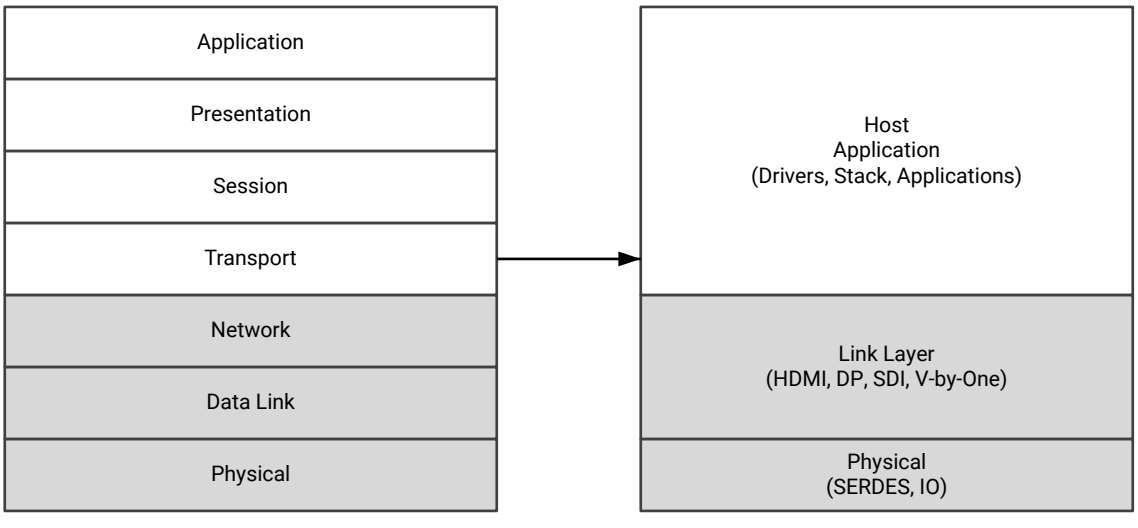
Core Overview

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.

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 the 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, use, and any limitations of the transceivers. For more information, see *UltraScale Architecture GTH Transceivers User Guide* (UG576) and *UltraScale Architecture GTY Transceivers User Guide* (UG578).

The following figure shows the standard OSI Model and mapping it with video IP solutions.

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

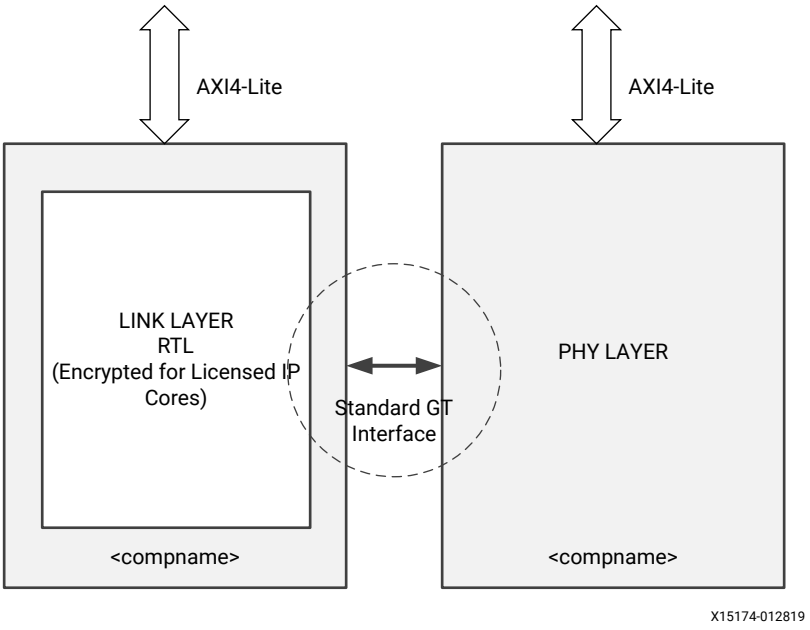


In accordance with the OSI model, the major PHY component for video IP cores is SerDes. Standardizing the SerDes delivery model provides benefits and flexibility for a video MAC layer at the system level.

The following figure 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: **Video IP Layer**



Applications

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

Unsupported Features

The following features of the standard are not supported in the core:

- Multi-MAC controllers support (complex use cases).

- Mixed MAC controller support, that is, HDMI on the input and DisplayPort output and so on. The current Video PHY Controller core supports the same protocol MAC on both the input and output.
- Multiple protocols per instance (for example, two HDMIs in one Video PHY Controller).
- Standalone use; it is designed to be used with the Xilinx® HDMI or DisplayPort MAC subsystems.

Licensing and Ordering

This Xilinx® LogiCORE™ IP module is provided under the terms of the Xilinx Core License Agreement. The module is shipped as part of the Vivado® Design Suite. For full access to all core functionalities in simulation and in hardware, you must purchase a license for the core. To generate a full license, visit the product licensing web page. Evaluation licenses and hardware timeout licenses might be available for this core. Contact your local Xilinx sales representative for information about pricing and availability.

***Note*:** To verify that you need a license, check the License column of the IP Catalog. Included means that a license is included with the Vivado® Design Suite; Purchase means that you have to purchase a license to use the core.

For more information about this core, 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 about pricing and availability of other Xilinx LogiCORE IP modules and tools, contact your local Xilinx sales representative.

License Checkers

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

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

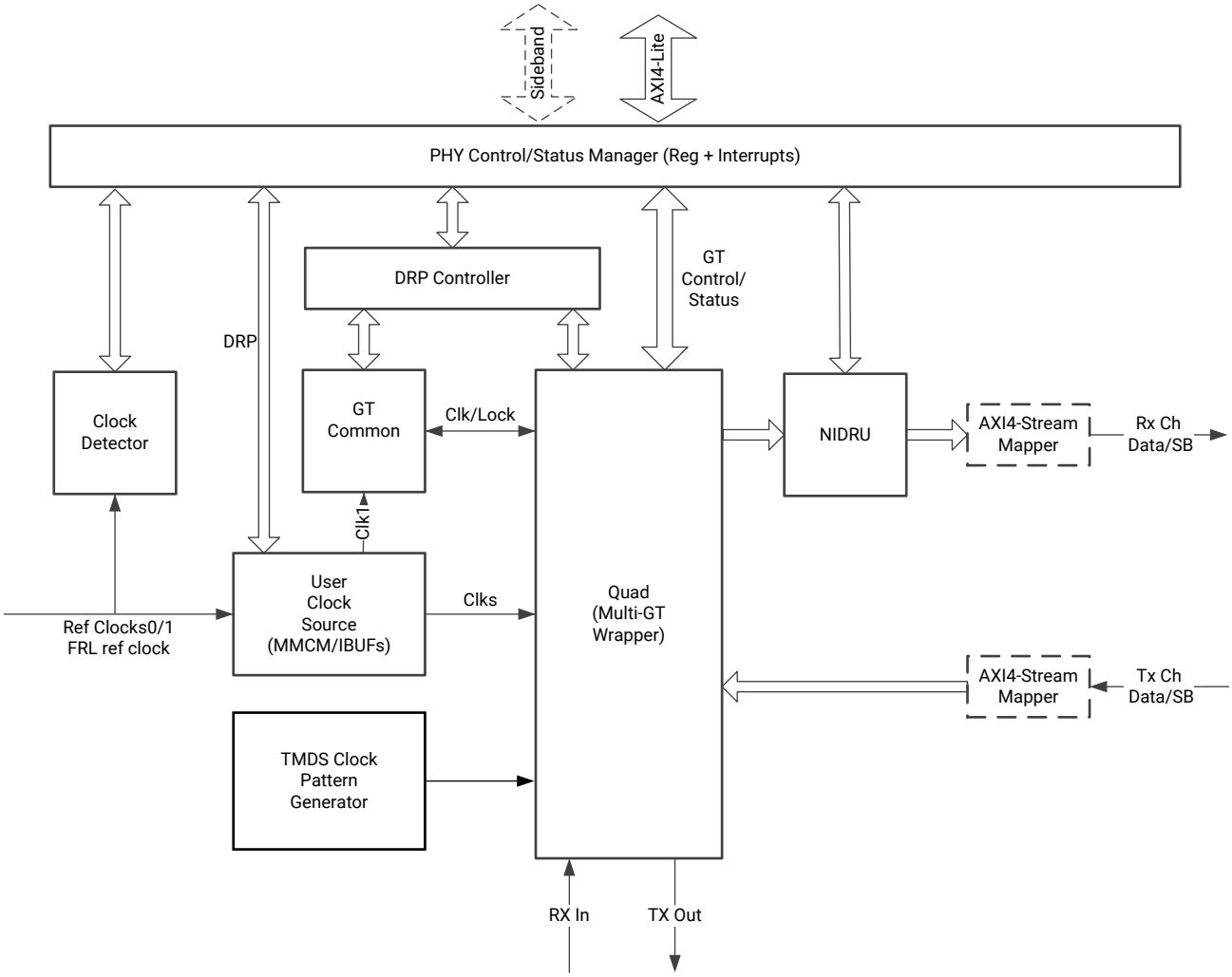
IMPORTANT! *IP license level is ignored at checkpoints. The test confirms a valid license exists. It does not check IP license level.*

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) use by providing a standardized interface and software programmability of serial transceiver functions.

The functional block diagram of the core is shown in the following figure.

Figure 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 and MMCM/PLL 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 cases where the TX buffer is bypassed, a mixed-mode clock manager (MMCM) generates the required output clocks based on TX/RXOUTCLK. In HDMI, along with generating USRCLK and USRCLK2, this block also produces video clocks and differential and single-ended TX Transition Minimized Differential Signaling (TMDS) CLK as per requirement of the *HDMI 1.4/2.0 Transmitter Subsystem Product Guide* (PG235) and *HDMI 1.4/2.0 Receiver Subsystem Product Guide* (PG236). 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 more information on HDMI clocking requirements, see the Clocking sections of the *HDMI 1.4/2.0 Transmitter Subsystem Product Guide* (PG235) and the *HDMI 1.4/2.0 Receiver Subsystem Product Guide* (PG236).
- **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 Thresholds
    - QPLL = 74.125 MHz
    - CPLL = 80.000 MHz
  - GTPE2 Thresholds
    - PLL0/1 = 80.000 MHz
  - GTHE3, GTHE4, and GTYE4 Thresholds:
    - QPLL0 = 61.250 MHz
    - CPLL = 50.00 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 HDMI Reference Clock Requirements. The NI-DRU reference clock can be same as the FRL reference clock.

- **TMDS Clock Pattern Generator:** This block is enabled when the Use 4th GT Channel as TX TMDS clock option is enabled. This block connects to the fourth GT channel and provides the pattern to transmit the TMDS clock through the GT channel.

Related Information

HDMI Reference Clock Requirements

Performance

The Video PHY Controller is designed to operate in coordination with the performance characteristics of the transceiver primitives it instantiates.

- For the DisplayPort 1.2 protocol, a 2-byte and 4-byte internal datapath are configured.
- For the DisplayPort 1.4 protocol, only a 2-byte internal datapath is supported.

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.

- *Zynq UltraScale+ RFSoC Data Sheet: DC and AC Switching Characteristics* (DS926)
- *Zynq UltraScale+ MPSoC Data Sheet: DC and AC Switching Characteristics* (DS925)
- *Kintex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics* (DS922)
- *Virtex UltraScale+ FPGA Data Sheet: DC and AC Switching Characteristics* (DS923)
- *Virtex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS893)
- *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS892)
- *Zynq-7000 SoC (Z-7030, Z-7035, Z-7045, and Z-7100) Data Sheet: DC and AC Switching Characteristics* (DS191)
- *Kintex-7 FPGAs Data Sheet: DC and AC Switching Characteristics* (DS182)
- *Virtex-7 FPGAs Data Sheet: DC and AC Switching Characteristics* (DS183)
- *Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics* (DS181)

Resource Use

For full details about performance and resource use, visit the Performance and Resource Use web page.

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 use 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 use results that 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

Clocking and Reset Interface Ports

Table 1: **Clocking and Resets Ports**

Name	I/O	Clock Domain	Description
mgtrefclk0_pad_(p/n)_in	I		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	I		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	I		Available when Advanced Clock mode is enabled. You must instantiate input buffers at the system level. 7 series: Connects to GTREFCLK0.

Table 1: **Clocking and Resets Ports** *(cont'd)*

Name	I/O	Clock Domain	Description
mgtrefclk1_in	I		Available when Advanced Clock mode is enabled. You must instantiate input buffers at the system level. 7 series: Connects to GTREFCLK1.
mgtrefclk(0/1)_odiv2_in	I		Available when Advanced Clock mode is enabled and GTREFCLK0/1 is selected as one of the input clocks. This must be connected to the ODIV2 output of the gtrefclk0/1_in input buffer. The ODIV2 output must be configured to output a divided-by-1 clock.
gtnorthrefclk(0/1)_in	I		Available when Advanced Clock mode is enabled (DisplayPort) 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.
gtsouthrefclk(0/1)_in	I		Available when Advanced Clock mode is enabled (DisplayPort) 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)	I		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)	I		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[1]	I		Available when Advanced Clock mode is enabled (DisplayPort) 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[1]	I		Available when Advanced Clock mode is enabled (DisplayPort) 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[1]	I		Available when Advanced Clock mode is enabled (DisplayPort) 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[1]	I		. You must instantiate input buffers at the system level. Connects to GTNORTHREFCLK11.
gtsouthrefclk00_in[1]	I		Available when Advanced Clock mode is enabled (DisplayPort) 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[1]	I		Available when Advanced Clock mode is enabled (DisplayPort) 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.

Table 1: **Clocking and Resets Ports** *(cont'd)*

Name	I/O	Clock Domain	Description
gtsouthrefclk10_in[1]	I		Available whenAvailable when Advanced Clock mode is enabled (DisplayPort) 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[1]	I		Available when Advanced Clock mode is enabled (DisplayPort) 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)	I		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)	I		Available when GTWESTREFCLK0/1 is selected as one of the input clock sources. Connects to GTWESTREFCLK0/1 ports.
drpclk[1]	I		Free running clock that is used to bring up the UltraScale device GT and to clock GT helper blocks.
vid_phy_tx_axi4s_aclk	I		Transmit AXI4-Stream Link Data interface clock
vid_phy_tx_axi4s_aresetn	I	TXUSRCLK2	Transmit AXI4-Stream Link Data interface reset. Unused port. It can be tied High or left unconnected.
vid_phy_rx_axi4s_aclk	I		Receive AXI4-Stream Link Data interface clock.
vid_phy_rx_axi4s_aresetn	I	RXUSRCLK2	Receive AXI4-Stream Link Data interface reset. Unused port. It can be tied High or left unconnected.
vid_phy_sb_aclk	I		Sideband interface clock. Connect to AXI4-Lite clock.
vid_phy_sb_aresetn	I	vid_phy_sb_aclk (AXI4-Lite)	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	O		Buffered clock sent out for programmable logic. Available in HDMI when TX is enabled.
rxoutclk	O		Buffered clock sent out for programmable logic. Available in HDMI when RX is enabled.
vid_phy_axi4lite_aclk	I		AXI Bus clock.
vid_phy_axi4lite_aresetn	I	AXI4-Lite	AXI reset. active-Low. Must be connected to ARESETN that is synched to vid_phy_axi4lite_aclk port (that is, peripheral_aresetn port of Processor System Reset IP)
tx_refclk_rdy[2]	I	Async	Active-High (default): 1- Locked 0 - Unlocked TX reference clock ready or lock indicator. See the HDMI Reference Clock Requirements (link below) for details about the tx_refclk_rdy port implementation. Active level is controlled by Tx RefClk Rdy active Vivado® IDE parameter. If set to Low, 1 - Unlocked 0 - Locked

Table 1: **Clocking and Resets Ports** *(cont'd)*

Name	I/O	Clock Domain	Description
tx_tmds_clk[2]	O		TX TMDS clock.
tx_tmds_clk_p/n	O		Differential TX TMDS clock output. ***Note*:** These ports are disabled when Vivado IDE option "Use 4th GT Channel as TX TMDS Clock" is checked.
tx_video_clk[2]	O		TX video clock.
txrefclk_ceb[2]	O	AXI4-Lite	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[2]	O		RX TMDS clock.
rx_tmds_clk_p/n[2]	O		Differential RX TMDS clock output.
rx_video_clk[2]	O		RX Video clock.
rxrefclk_ceb[2]	O	AXI4-Lite	RX external reference clock IBUFDS CEB. Available when RX selects reference clock source from NORTH, SOUTH, EAST, or WEST. Example GTNORTHREFCLK_0/1.

Notes:

1. Used for UltraScale and UltraScale+ devices.
2. Used for HDMI.

Related Information

HDMI Reference Clock Requirements

GT Channels Interface Ports

Table 2: **GT Channels Ports**

Name	I/O	Clock Domain	Description
gttxpippmen_in[1]	I	TXUSRCLK2	Width: 1* Num. channels. Available when TXPI_Port_EN user parameter is enabled. Connects to TXPIPPMEN on transceiver channel primitives.
gttxpippmovrden_in[1]	I	TXUSRCLK2	Width: 1* Num. channels. Available when TXPI_Port_EN user parameter is enabled. Connects to TXPIPPMOVRDEN on transceiver channel primitives.
gttxpippmpd_in[1]	I	TXUSRCLK2	Width: 1* Num. channels. Available when TXPI_Port_EN user parameter is enabled. Connects to TXPIPPMPD on transceiver channel primitives.
gttxpippmsel_in[1]	I	TXUSRCLK2	Width: 1* Num. channels. Available when TXPI_Port_EN user parameter is enabled. Connects to TXPIPPMSEL on transceiver channel primitives.
gttxpippmstepsize_in[1]	I	TXUSRCLK2	Width: 5* Num. channels. Available when TXPI_Port_EN user parameter is enabled. Connects to TXPIPPMSTEPSIZE on transceiver channel primitives.
phy_rxp_in[n-1:0][2]	I	RX Serial Clock	Positive differential serial input to the transceiver

Table 2: **GT Channels Ports** *(cont'd)*

Name	I/O	Clock Domain	Description
phy_rxn_in[n-1:0][2]	I	RX Serial Clock	Negative differential serial input to the transceiver
phy_txp_out[n-1:0][3]	O	TX Serial Clock	Positive differential serial output from the transceiver
phy_txn_out[n-1:0][3]	O	TX Serial Clock	Negative differential serial output from the transceiver
vid_phy_tx_axi4s_ch<i>_tready[4]	O	TXUSRCLK2	AXI4-Stream based tready indicator
vid_phy_tx_axi4s_ch<i>_tvalid[4]	I	TXUSRCLK2	AXI4-Stream based tvalid indicator
vid_phy_tx_axi4s_ch<i>_tdata[4]	I	TXUSRCLK2	AXI4-Stream based tdata bus GT Mapping: TXDATA_IN Width: TX_DATA_WIDTH
vid_phy_tx_axi4s_ch<i>_tuser[4]	I	TXUSRCLK2	AXI4-Stream based tuser bus GT Mapping: {TXCHARDISPVAL, TXCHARDISPMODE, TXCHARISK} In UltraScale and UltraScale+ devices, TXCHARDISPVAL and TXCHARDISPMODE are represented using TXCTLR0 and TXCTRL1. Width: TX_USER_WIDTH (For DisplayPort 12 bits)
vid_phy_rx_axi4s_ch<i>_tready[4]	I	RXUSRCLK2	AXI4-Stream based tready indicator
vid_phy_rx_axi4s_ch<i>_tvalid[4]	O	RXUSRCLK2	AXI4-Stream based tvalid indicator.
vid_phy_rx_axi4s_ch<i>_tdata[4]	O	RXUSRCLK2	AXI4-Stream based tdata bus GT Mapping: RXDATAOUT Width: RX_DATA_WIDTH
vid_phy_rx_axi4s_ch<i>_tuser[4]	O	RXUSRCLK2	AXI4-Stream based tuser bus GT Mapping: {RXNOTINTABLE, RXDISPERR, RXCHARISK} In UltraScale and UltraScale+ devices, RXNOTINTABLE, RXDISPERR, RXCHARISK are represented using RXCTRL0, RXCTR1 and RXCTRL3. • bit 0 - RXCHARISK or RXCTRL2 (UltraScale and UltraScale +) • bit 1 - RXDISPERR or RXCTRL1 (UltraScale and UltraScale +) • bit 2 - RXNOTINTABLE or RXCTRL0(UltraScale and UltraScale+) Width: RX_USER_WIDTH (For DisplayPort: 12 bits)

Notes:

1. Used for UltraScale and UltraScale+ devices.
2. n is the number of RX channels.
3. n is the number of TX channels.
4. <i> is the transceiver channel index.

Sideband Signals Interface Ports (Optional)

Table 3: **Sideband Signals Optional Ports**

Name[1]	I/O	Description
vid_phy_control_sb_tx_tready	O	AXI4-Stream based tready indicator
vid_phy_control_sb_tx_tdata	I	AXI4-Stream based tdata bus Not used by the DisplayPort Protocol
vid_phy_control_sb_tx_tvalid	I	AXI4-Stream based tvalid
vid_phy_status_sb_tx_tready	I	AXI4-Stream based tready indicator
vid_phy_status_sb_tx_tdata[7:0] (DisplayPort) vid_phy_status_sb_tx_tdata[1:0] (HDMI)	O	AXI4-Stream based tdata bus For DisplayPort, see DisplayPort Transmit - Status Path. For HDMI™, see HDMI Transmit - Status Path.
vid_phy_status_sb_tx_tvalid	O	AXI4-Stream based tvalid
vid_phy_control_sb_rx_tready	O	AXI4-Stream based tready indicator
vid_phy_control_sb_rx_tdata[7:0] (DisplayPort Only)	I	AXI4-Stream based tdata bus. For DisplayPort, see DisplayPort Receive - Control Path.
vid_phy_control_sb_rx_tvalid (DisplayPort Only)	I	AXI4-Stream based tvalid
vid_phy_status_sb_rx_tready (DisplayPort Only)	I	AXI4-Stream based tready indicator
vid_phy_status_sb_rx_tdata{15:0] (DisplayPort) vid_phy_status_sb_rx_tdata[1:0] (HDMI)	O	AXI4-Stream based tdata bus. For DisplayPort, see DisplayPort Receive - Status Path. For HDMI, see HDMI Receive - Status Path.
vid_phy_status_sb_rx_tvalid	O	AXI4-Stream based tvalid

Notes:

1. Clock Domain: Sideband clock.

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 4: **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

Table 4: **DisplayPort Transmit Status Sideband Definition** *(cont'd)*

Bit Position	Status Details
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 5: **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 MMCM 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 6: **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 7: **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

Table 7: **Receive Status Sideband Definition** *(cont'd)*

Bit Position	Status Details
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
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 8: **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 MMCM block is stable.

AXI4-Lite Ports

The clock domain for the following table is the AXI4-Lite clock.

Table 9: **AXI4-Lite Ports**

Name	I/O	Description
vid_phy_axi4lite_awaddr[9:0]	I	Write address
vid_phy_axi4lite_awprot[2:0]	I	Protection type
vid_phy_axi4lite_awvalid	I	Write address valid
vid_phy_axi4lite_awready	O	Write address ready
vid_phy_axi4lite_awdata[31:0]	I	Write data bus
vid_phy_axi4lite_awstrb[3:0]	I	Write strobes
vid_phy_axi4lite_wvalid	I	Write valid
vid_phy_axi4lite_wready	O	Write ready
vid_phy_axi4lite_bresp[1:0]	O	Write response
vid_phy_axi4lite_bvalid	O	Write response valid
vid_phy_axi4lite_bready	I	Response ready

Table 9: **AXI4-Lite Ports** *(cont'd)*

Name	I/O	Description
vid_phy_axi4lite_araddr[9:0]	I	Read address
vid_phy_axi4lite_arprot[2:0]	I	Protection type
vid_phy_axi4lite_arvalid	I	Read address valid
vid_phy_axi4lite_arready	O	Read address ready
vid_phy_axi4lite_rdata[31:0]	O	Read data
vid_phy_axi4lite_rresp[1:0]	O	Read response
vid_phy_axi4lite_rvalid[1:0]	O	Read valid
vid_phy_axi4lite_rready	I	Read ready
irq	O	Interrupt output

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 the following register tables 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 the following tables, and the base address is configured by the AXI interconnect at the system level.

For more information, see the following documents:

- *7 Series FPGAs GTX/GTH Transceivers User Guide* (UG476)
- *UltraScale Architecture GTH Transceivers User Guide* (UG576)
- *7 Series FPGAs GTP Transceivers User Guide* (UG482)
- *UltraScale Architecture GTY Transceivers User Guide* (UG578)

Table 10: **Register Address Space**

Address (hex)	Register
0x0000	Version Register (VR)
0x0004	Reserved
0x0008	Reserved
0x000C	Bank Select (BSR)
Shared Features and Resets	
0x0010	Reference Clock Selection (RCS)
0x0014	PLL Reset (PR)

Table 10: **Register Address Space** *(cont'd)*

Address (hex)	Register
0x0018	PLL Lock Status (PLS)
0x001C	TX Initialization (TXI)
0x0020	TX Initialization Status (TXIS)
0x0024	RX Initialization (RXI)
0x0028	RX Initialization Status (RXIS)
0x002C	IBUFDS GTxx Control (IBUFDSGTxxCTRL)
Power Down	
0x0030	Power Down Control (PDC)
0x0034	Reserved
Loopback	
0x0038	Loopback Control (LBC)
0x003C	Reserved
Dynamic Reconfiguration Port (DRP)	
0x0040	DRP CONTROL Channel1 (DRPCCH1)
0x0044	DRP CONTROL Channel2 (DRPCCH2)
0x0048	DRP CONTROL Channel3 (DRPCCH3)
0x004C	DRP CONTROL Channel4 (DRPCCH4)
0x0050	DRP STATUS Channel1 (DRPSCH1)
0x0054	DRP STATUS Channel2 (DRPSCH2)
0x0058	DRP STATUS Channel3 (DRPSCH3)
0x005C	DRP STATUS Channel4 (DRPSCH4)
0x0060	DRP CONTROL Common (DRPCC)
0x0064	DRP STATUS Common (DRPSC)
0x0068 - 0x006C	Reserved
Transmitter Functions	
0x0070	TX Control (TXC)
0x0074[1]	TX Buffer Bypass Control (TXBBC)
0x0078	TX Status (TXS)
0x007C	TX DRIVER Control – Channel 1 and 2 (TXDC12)
0x0080	TX DRIVER Control – Channel 3 and 4 (TXDC34)
0x0084	RX Symbol Error Counter – Channel 1 and 2
0x0088	RX Symbol Error Counter – Channel 3 and 4
0x008C to 0x009C	Reserved
Receiver Functions	
0x0100	RX Control (RXC)
0x0104	RX Status (RXS)
0x0108	RX Equalization and CDR
0x010C[2]	RX TDLOCK VALUE

Table 10: **Register Address Space** *(cont'd)*

Address (hex)	Register
Interrupts Registers	
0x0110	Interrupt Enable Register (IER)
0x0114	Interrupt Disable Register (IDR)
0x0118	Interrupt Mask Register (IMR)
0x011C	Interrupt Status Register (ISR)
TXUSRCLK Clocking	
0x0120	MMCM TXUSRCLK Control/Status (MMCM_TXUSRCLK_CTRL)
0x0124	DRP CONTROL MMCM TXUSRCLK
0x0128	DRP STATUS MMCM TXUSRCLK
0x0134	BUFGGT TXUSRCLK Control (BUFGGT_TXUSRCLK_CTRL)
0x0138	MISC TXUSRCLK Control (MISC_TXUSRCLK_CTRL)
0x0140	MMCM RXUSRCLK Control/Status (MMCM_RXUSRCLK_CTRL)
0x0144	DRP CONTROL MMCM RXUSRCLK
0x0148	DRP STATUS MMCM RXUSRCLK
0x0154	BUFGGT RXUSRCLK Control (BUFGGT_RXUSRCLK_CTRL)
0x0158	MISC RXUSRCLK Control (MISC_RXUSRCLK_CTRL)
Clock Detector (HDMI)	
0x0200	Control Register
0x204	Status Register
0x0208	Frequency Counter Timeout
0x020C	Transmitter Frequency
0x0210	Receiver Frequency
0x0214	Transmitter Timer
0x0218	Receiver Timer
0x021C	DRU Frequency
Data Recovery Unit	
0x0300	Control Register
0x0304	Status Register
0x0308	Center Frequency Low Register – All Channels
0x030C	Center Frequency High Register – All Channels
0x0310	Gain Register – All Channels
TX TMDS Pattern Generator	
0x0340	Control Register

Notes:

1. For the DisplayPort protocol, the GT Wizards internal FSM handles these status bits. This register is unused.
2. For the DisplayPort protocol, this register field is unused.

Version Register (0x0000)

Table 11: **Version Register (VR)**

Bit	Default Value	Access Type	Description
31:24	0	RO	Core major version.
23:16	0	RO	Core minor version
15:12	0	RO	Core version revision
11:8	0	RO	Core patch details
7:0	0	RO	Internal revision

Notes:

1. For video_phy_controller_v2_0, VR is 32'h02_02_00_00.

Bank Select Register (BSR) (0x000C)

Table 12: **Bank Select Register**

Bit	Default Value	Access Type	Description
4:0	0	RW	TX Bank Selection
12:8	0	RW	RX Bank Selection

Shared Features and Reset Registers

Reference Clock Selection (RCS) Register (0x0010)

Table 13: **Reference Clock Selection Register**

Bit	Default Value	Access Type	Description
3:0	0	RW	QPLL (GTXE2) / QPLL0REFCLKSEL (GTHE3/GTHE4/GTYE4) / PLL0REFCLKSEL (GTPE2)
7:4	0	RW	CPLLREFCLKSEL
11:8	0	RW	QPLL1REFCLKSEL (For UltraScale/UltraScale+ devices)/PLL1REFCLKSEL (GTPE2)
23:12	0	RW	Reserved
27:24	0	RW	{TXSYSCLKSEL[1:0], RXSYSCLKSEL[1:0]}
31:28	0	RW	{TXPLLCLKSEL[1:0], RXPLLCLKSEL[1:0]} For UltraScale and UltraScale+ devices

PLL Reset Register (PR) (0x0014)

Note:

This is applicable only to the DisplayPort. It is not used in HDMI.

Table 14: **PLL Reset Register**

Bit	Default Value	Access Type	Description
0	0	RW	CPLLRESET
1	0	RW	QPLL0RESET
2	0	RW	QPLL1RESET – For UltraScale and UltraScale+ devices

PLL Lock Status (PLS) Register (0x0018)

Table 15: **PLL Lock Status Register**

Bit	Default Value	Access Type	Description
0	0	RO	CPLLLOCK—Ch1
1	0	RO	CPLLLOCK—Ch2
2	0	RO	CPLLLOCK—Ch3
3	0	RO	CPLLLOCK—Ch4
4	0	RO	QPLL/QPLL0LOCK/PLL0LOCK
5	0	RO	QPLL1LOCK – For UltraScale/UltraScale+ devices PLL1LOCK

TX Initialization (TXI) Register (0x001C)

Table 16: **TX Initialization Register**

Bit	Default Value	Access Type	Description
Channel 1			
0	0	RW	GTTXRESET
1	0	RW	TXPMARESET
2	0	RW	TXPCSRESET
3	0	RW	TXUSERRDY
6:4	0	RW	Reserved
7	0	RW	PLL_GT_RESET
Channel 2			
8	0	RW	GTTXRESET
9	0	RW	TXPMARESET
10	0	RW	TXPCSRESET
11	0	RW	TXUSERRDY
14:12	0	RW	Reserved
15	0	RW	PLL_GT_RESET
Channel 3			
16	0	RW	GTTXRESET

Table 16: **TX Initialization Register** *(cont'd)*

Bit	Default Value	Access Type	Description
17	0	RW	TXPMARESET
18	0	RW	TXPCSRESET
19	0	RW	TXUSERRDY
22:20	0	RW	Reserved
23	0	RW	PLL_GT_RESET
Channel 4			
24	0	RW	GTTXRESET
25	0	RW	TXPMARESET
26	0	RW	TXPCSRESET
27	0	RW	TXUSERRDY
30:28	0	RW	Reserved
31	0	RW	PLL_GT_RESET

TX Initialization Status (TXIS) Register (0x0020)

Table 17: **TX Initialization Status (TXIS)**

Bit	Default Value	Access Type	Description
Channel 1			
0	0	RO	TXRESETDONE
1	0	RO	TXPMARESETDONE[1]
2	0	RO	GTPOWERGOOD (For UltraScale+ devices)[2]
7:3	0	RO	Reserved
Channel 2			
8	0	RO	TXRESETDONE
9	0	RO	TXPMARESETDONE[1]
10	0	RO	GTPOWERGOOD (For UltraScale+ devices)[2]
15:11	0	RO	Reserved
Channel 3			
16	0	RO	TXRESETDONE
17	0	RO	TXPMARESETDONE[1]
18	0	RO	GTPOWERGOOD (For UltraScale+ devices)[2]
23:19	0	RO	Reserved
Channel 4			
24	0	RO	TXRESETDONE
25	0	RO	TXPMARESETDONE[1]
26	0	RO	GTPOWERGOOD (For UltraScale+ devices)

Table 17: **TX Initialization Status (TXIS)** *(cont'd)*

Bit	Default Value	Access Type	Description
31:27	0	RO	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. The signals driving the GTPOWERGOOD in the TXIS and RXIS registers are identical.

RX Initialization (RXI) Register (0x0024)

Table 18: **RX Initialization (RXI) Register**

Bit	Default Value	Access Type	Description
Channel 1			
0	0	RW	GTRXRESET
1	0	RW	RXPMARESET
2	0	RW	RXDFELPMRESET[1]
3	0	RW	RESERVED[2]
4	0	RW	RXPCSRESET[2]
5	0	RW	RXBUFRESET
6	0	RW	RXUSERRDY[3]
7	0	RW	PLL_GT_RESET
Channel 2			
8	0	RW	GTRXRESET
9	0	RW	RXPMARESET
10	0	RW	RXDFELPMRESET[1]
11	0	RW	RESERVED[2]
12	0	RW	RXPCSRESET[2]
13	0	RW	RXBUFRESET
14	0	RW	RXUSERRDY[3]
15	0	RW	PLL_GT_RESET
Channel 3			
16	0	RW	GTRXRESET
17	0	RW	RXPMARESET
18	0	RW	RXDFELPMRESET[1]
19	0	RW	RESERVED[2]
20	0	RW	RXPCSRESET[2]
21	0	RW	RXBUFRESET
22	0	RW	RXUSERRDY[3]
23	0	RW	PLL_GT_RESET

Table 18: **RX Initialization (RXI) Register** *(cont'd)*

Bit	Default Value	Access Type	Description
Channel 4			
24	0	RW	GTRXRESET
25	0	RW	RXPMARESET
26	0	RW	RXDFELPMRESET[1]
27	0	RW	RESERVED[2]
28	0	RW	RXPCSRESET[2]
29	0	RW	RXBUFRESET
30	0	RW	RXUSERRDY[3]
31	0	RW	PLL_GT_RESET

Notes:

1. For the DisplayPort protocol, RXDFELPMRESET is an unused field.
2. For the DisplayPort protocol, this register field is unused for UltraScale and UltraScale+ devices.
3. For the DisplayPort protocol, this register field is unused for UltraScale and UltraScale+ devices. The internal GT UltraScale wizard FSM handles this field.

RX Initialization Status (RXIS) Register (0x0028)

Table 19: **RX Initialization Status (RXIS) Register**

Bit	Default Value	Access Type	Description
Channel 1			
0	0	RO	RXRESETDONE
1	0	RO	RXPMARESETDONE[1]
2	0	RO	GTPOWERGOOD (For UltraScale+ devices)[2]
7:3	0	RO	Reserved
Channel 2			
8	0	RO	RXRESETDONE
9	0	RO	RXPMARESETDONE[1]
10	0	RO	GTPOWERGOOD (For UltraScale+ devices)[2]
15:11	0	RO	Reserved
Channel 3			
16	0	RO	RXRESETDONE
17	0	RO	RXPMARESETDONE[1]
18	0	RO	GTPOWERGOOD (For UltraScale+ devices)[2]
23:19	0	RO	Reserved
Channel 4			
24	0	RO	RXRESETDONE
25	0	RO	RXPMARESETDONE[1]

Table 19: **RX Initialization Status (RXIS) Register** *(cont'd)*

Bit	Default Value	Access Type	Description
26	0	RO	GTPOWERGOOD (For UltraScale+ devices)[2]
31:27	0	RO	Reserved

Notes:

1. For the DisplayPort protocol, RXPMARESETDONE is used only for UltraScale and UltraScale+ devices. For 7 series devices, this field is tied to 1.
2. The signals driving the GTPOWERGOOD in the TXIS and RXIS registers are identical.

IBUFDS GTxx Control (IBUFDSGTxxCTRL) Register (0x002C)

This control is used manage the CEB pin of IBUFDS_GTE2/3/4 based on the device. Low (0) to enable; High (1) to disable.

Table 20: **IBUFDS GTxx Control (IBUFDSGTxxCTRL) Register**

Bit	Default Value	Access Type	Description
0	0	RW	GTREFCLK0_CEB
1	0	RW	GTREFCLK1_CEB

Power Down Control (PDC) Register (0x0030)

Table 21: **Power Down Control (PDC) Register**

Bit	Default Value	Access Type	Description
Channel 1			
0	0	RW	CPLLPD[1]
1	0	RW	QPLL0PD/PLL0PD
2	0	RW	QPLL1PD (For UltraScale and UltraScale+ devices)/PLL1PD
4:3	0	RW	RXPD[1:0]
6:5	0	RW	TXPD[1:0]
7	N/A	N/A	Reserved
Channel 2			
8	0	RW	CPLLPD[1]
10:9	N/A	N/A	Reserved
12:11	0	RW	RXPD[1:0]
14:13	0	RW	TXPD[1:0]
15	N/A	N/A	Reserved
Channel 3			
16	0	RW	CPLLPD[1]

Table 21: **Power Down Control (PDC) Register** *(cont'd)*

Bit	Default Value	Access Type	Description
18:17	N/A	N/A	Reserved
20:19	0	RW	RXPD[1:0]
22:21	0	RW	TXPD[1:0]
23	N/A	N/A	Reserved
Channel 4			
24	0	RW	CPLLPD[1]
26:25	N/A	N/A	Reserved
28:27	0	RW	RXPD[1:0]
30:29	0	RW	TXPD[1:0]
31	N/A	N/A	Reserved

Notes:

1. This field is applicable only for 7 series devices. For UltraScale and UltraScale+ devices, CPLLPD is handled by the GT wizard FSMs internally.

Loopback Control (LBC) Register (0x0038)

Table 22: **Loopback Control Register**

Bit	Default Value	Access Type	Description
Channel 1			
2:0	0	RW	LOOPBACK[2:0][1]
7:3	0	RW	Reserved
Channel 2			
10:8	0	RW	LOOPBACK[2:0][1]
15:11	0	RW	Reserved
Channel 3			
18:16	0	RW	LOOPBACK[2:0][1]
23:19	0	RW	Reserved
Channel 4			
26:24	0	RW	LOOPBACK[2:0][1]
31:27	0	RW	Reserved

Notes:

1. Only Near-end PMA Loopback (010b) and Normal Operation (000b) are supported.

Dynamic Reconfiguration Port (DRP) Registers

Dynamic Control Channel 1 (DRPCCH1) Register (0x0040)

Table 23: **Dynamic Control Channel 1 Register**

Bit	Default Value	Access Type	Description
11:0	0	RW	DRPADDR[8:0]
12	0	RW	DRPEN
13	0	RW	DRPWE
14	0	RW	DRP Reset (For UltraScale and UltraScale+ devices)[1]
15	0	RW	Reserved
31:16	0	RW	DRPDI [15:0]

Notes:

1. For the DisplayPort protocol, this register field is unused.

Dynamic Control Channel 2 (DRPCCH2) Register (0x0044)

Table 24: **Dynamic Control Channel 2 Register**

Bit	Default Value	Access Type	Description
11:0	0	RW	DRPADDR[8:0]
12	0	RW	DRPEN
13	0	RW	DRPWE
14	0	RW	DRP Reset (For UltraScale and UltraScale+ devices)[1]
15	0	RW	Reserved
31:16	0	RW	DRPDI [15:0]

Notes:

1. For the DisplayPort protocol, this register field is unused.

Dynamic Control Channel 3 (DRPCCH3) Register (0x0048)

Table 25: **Dynamic Control Channel 3 Register**

Bit	Default Value	Access Type	Description
11:0	0	RW	DRPADDR[8:0]
12	0	RW	DRPEN
13	0	RW	DRPWE
14	0	RW	DRP Reset (For UltraScale and UltraScale+ devices)[1]
15	0	RW	Reserved

Table 25: **Dynamic Control Channel 3 Register** *(cont'd)*

Bit	Default Value	Access Type	Description
31:16	0	RW	DRPDI [15:0]

Notes:
1. For the DisplayPort protocol, this register field is unused.

Dynamic Control Channel 4 (DRPCCH4) Register (0x004C)

Table 26: **Dynamic Control Channel 4 Register**

Bit	Default Value	Access Type	Description
11:0	0	RW	DRPADDR[8:0]
12	0	RW	DRPEN
13	0	RW	DRPWE
14	0	RW	DRP Reset (For UltraScale and UltraScale+ devices)[1]
15	0	RW	Reserved
31:16	0	RW	DRPDI [15:0]

Notes:
1. For the DisplayPort protocol, this register field is unused.

DRP Status Channel 1 (DRPSCH1) Register (0x0050)

Table 27: **DRP Status Channel 1 Register**

Bit	Default Value	Access Type	Description
15:0	0	RO	DRPDO[15:0] – Valid for Read Transfers
16	0	RO	DRPRDY (Indicates valid transfer)
17	0	RO	DRPBUSY (Indicated DRP port free)
31:18	0	RO	Reserved

DRP Status Channel 2 (DRPSCH2) Register (0x0054)

Table 28: **DRP Status Channel 2 Register**

Bit	Default Value	Access Type	Description
15:0	0	RO	DRPDO[15:0] – Valid for Read Transfers
16	0	RO	DRPRDY (Indicates valid transfer)
17	0	RO	DRPBUSY (Indicated DRP port free)
31:18	0	RO	Reserved

DRP Status Channel 3 (DRPSCH3) Register (0x0058)

Table 29: **DRP Status Channel 3 Register**

Bit	Default Value	Access Type	Description
15:0	0	RO	DRPDO[15:0] – Valid for Read Transfers
16	0	RO	DRPRDY (Indicates valid transfer)
17	0	RO	DRPBUSY (Indicated DRP port free)
31:18	0	RO	Reserved

DRP Status Channel 4 (DRPSCH4) Register (0x005C)

Table 30: **DRP Status Channel 4 Register**

Bit	Default Value	Access Type	Description
15:0	0	RO	DRPDO[15:0] – Valid for Read Transfers
16	0	RO	DRPRDY (Indicates valid transfer)
17	0	RO	DRPBUSY (Indicated DRP port free)
31:18	0	RO	Reserved

DRP Control Common (DRPCC) Register (0x0060)

Table 31: **DRP Control Common Register**

Bit	Default Value	Access Type	Description
11:0	0	RW	DRPADDR[8:0] (8 bits for 7 series)
12	0	RW	DRPEN
13	0	RW	DRPWE
15:14	0	RW	Reserved
31:16	0	RW	DRPDI [15:0]

DRP Status Common (DRPSC) Register (0x0064)

Table 32: **DRP Status Common Register**

Bit	Default Value	Access Type	Description
15:0	0	RO	DRPDO[15:0] – Valid for Read Transfers
16	0	RO	DRPRDY (Indicates valid transfer)
17	0	RO	DRP BUSY (Indicated DRP port free)
31:18	0	RO	Reserved

Transmitter Functions Registers

TX Control (TXC) Register (0x0070)

Table 33: **TX Control (TXC) Register**

Bit	Default Value	Access Type	Description
Channel 1			
0	0	RW	TX8B10BEN[1]
1	0	RW	TXPOLARITY (0: Not-inverted 1: Inverted)
4:2	0	RW	TXPRBSSEL[2:0] 7 series PRBS modes: 000 - Standard operation 001 - PRBS-7 010 - PRBS-15 011 - PRBS-23 100 - PRBS-31 UltraScale /UltraScale+ PRBS modes: 000 - Standard operation 001 - PRBS-7 010 - PRBS-9 010 - PRBS-15 011 - PRBS-23 100 - PRBS-31
5	0	RW	TXPRBSFORCEERR
6	0	RW	TXPRBSSEL[2]
7	0	RW	Reserved
Channel 2			
8	0	RW	TX8B10BEN[1]
9	0	RW	TXPOLARITY (0: Not-inverted 1: Inverted)
12:10	0	RW	TXPRBSSEL[2:0]. Refer to Channel 1 for PRBS modes.
13	0	RW	TXPRBSFORCEERR
14	0	RW	TXPRBSSEL[2]
15	0	RW	Reserved
Channel 3			
16	0	RW	TX8B10BEN[1]
17	0	RW	TXPOLARITY (0: Not-inverted 1: Inverted)
20:18	0	RW	TXPRBSSEL[2:0]. Refer to Channel 1 for PRBS modes.
21	0		TXPRBSFORCEERR
22	0	RW	TXPRBSSEL[2]
23	0	RW	Reserved
Channel 4			
24	0	RW	TX8B10BEN[1]

Table 33: **TX Control (TXC) Register** *(cont'd)*

Bit	Default Value	Access Type	Description
25	0	RW	TXPOLARITY (0: Not-inverted 1: Inverted)
28:26	0	RW	TXPRBSSEL[2:0]. Refer to Channel 1 for PRBS modes.
29	0	RW	TXPRBSFORCEERR
30	0	RW	TXPRBSSEL[2]
31	0	RW	Reserved

Notes:

1. For the HDMI protocol, this register field is unused.
2. For the DisplayPort protocol, RXDFELPMRESET is an unused field.

TX Buffer Bypass Control (TXBBC) Register (0x0074)

For the DisplayPort protocol, the GT Wizards internal FSM handles these status bits. This register is unused.

Table 34: **TX Buffer Bypass Control Register**

Bit	Default Value	Access Type	Description
Channel 1			
0	0	RW	TXPHDLYRESET
1	0	RW	TXPHALIGN
2	0	RW	TXPHALIGNEN
3	0	RW	TXPHDLYPD
4	0	RW	TXPHINIT
5	0	RW	TXDLYRESET
6	0	RW	TXDLYBYPASS
7	0	RW	TXDLYEN
Channel 2			
8	0	RW	TXPHDLYRESET
9	0	RW	TXPHALIGN
10	0	RW	TXPHALIGNEN
11	0	RW	TXPHDLYPD
12	0	RW	TXPHINIT
13	0	RW	TXDLYRESET
14	0	RW	TXDLYBYPASS
15	0	RW	TXDLYEN
Channel 3			
16	0	RW	TXPHDLYRESET
17	0	RW	TXPHALIGN

Table 34: **TX Buffer Bypass Control Register** *(cont'd)*

Bit	Default Value	Access Type	Description
18	0	RW	TXPHALIGNEN
19	0	RW	TXPHDLYPD
20	0	RW	TXPHINIT
21	0	RW	TXDLYRESET
22	0	RW	TXDLYBYPASS
23	0	RW	TXDLYEN
Channel 4			
24	0	RW	TXPHDLYRESET
25	0	RW	TXPHALIGN
26	0	RW	TXPHALIGNEN
27	0	RW	TXPHDLYPD
28	0	RW	TXPHINIT
29	0	RW	TXDLYRESET
30	0	RW	TXDLYBYPASS
31	0	RW	TXDLYEN

TX Status (TXS) Register (0x0078)

Table 35: **TX Status (TXS) Register**

Bit	Default Value	Access Type	Description
Channel 1			
0	0	RW	TXPHALIGNDONE
1	0	RW	TXPHINITDONE[1]
2	0	RW	TXDLYRESETDONE[1]
4:3	0	RW	TXBUFSTATUS[1:0]
5	0	RW	TXBUFFBYPASS_ERROR (UltraScale and UltraScale+ devices)
7:6	0	RW	Reserved
Channel 2			
8	0	RW	TXPHALIGNDONE
9	0	RW	TXPHINITDONE[1]
10	0	RW	TXDLYRESETDONE[1]
12:11	0	RW	TXBUFSTATUS[1:0]
13	0	RW	TXBUFFBYPASS_ERROR (UltraScale and UltraScale+ devices)
15:14	0	RW	Reserved
Channel 3			
16	0	RW	TXPHALIGNDONE
17	0	RW	TXPHINITDONE[1]

Table 35: **TX Status (TXS) Register** *(cont'd)*

Bit	Default Value	Access Type	Description
18	0	RW	TXDLYRESETDONE[1]
20:19	0	RW	TXBUFSTATUS[1:0]
21	0	RW	TXBUFFBYPASS_ERROR (UltraScale and UltraScale+ devices)
23:22	0	RW	Reserved
Channel 4			
24	0	RW	TXPHALIGNDONE
25	0	RW	TXPHINITDONE[1]
26	0	RW	TXDLYRESETDONE[1]
28:27	0	RW	TXBUFSTATUS[1:0]
29	0	RW	TXBUFFBYPASS_ERROR (UltraScale and UltraScale+ devices)
31:30	0	RW	Reserved

Notes:

1. For the DisplayPort protocol, this register field is unused.

TX DRIVER Control – Channel 1 and 2 (TXDC12) Register (0x007C)

Table 36: **TX DRIVER Control – Channel 1 and 2 (TXDC12) Register**

Bit	Default Value	Access Type	Description
Channel 1			
3:0	0	RW	TXDIFFCTRL[3:0]
4	0	RW	TXELECIDLE[1]
5	0	RW	TXINHIBIT
10:6	0	RW	TXPOSTCURSOR[4:0]
15:11	0	RW	TXPRECURSOR[4:0]
Channel 2			
19:16	0	RW	TXDIFFCTRL [3:0]
20	0	RW	TXELECIDLE[1]
21	0	RW	TXINHIBIT
26:22	0	RW	TXPOSTCURSOR[4:0]
31:27	0	RW	TXPRECURSOR[4:0]

Notes:

1. For the DisplayPort protocol, this register field is unused.

TX DRIVER Control – Channel 3 and 4 (TXDC34) Register (0x0080)

Table 37: **TX DRIVER Control – Channel 3 and 4 (TXDC34) Register**

Bit	Default Value	Access Type	Description
Channel 3			
3:0	0	RW	TXDIFFCTRL[3:0]
4	0	RW	TXELECIDLE[1]
5	0	RW	TXINHIBIT
10:6	0	RW	TXPOSTCURSOR[4:0]
15:11	0	RW	TXPRECURSOR[4:0]
Channel 4			
19:16	0	RW	TXDIFFCTRL [3:0]
20	0	RW	TXELECIDLE[1]
21	0	RW	TXINHIBIT
26:22	0	RW	TXPOSTCURSOR[4:0]
31:27	0	RW	TXPRECURSOR[4:0]

Notes:
1. For the DisplayPort protocol, this register field is unused.

Receiver Functions

RX Control (RXC) Register (0x0100)

Table 38: **RX Control Register**

Bit	Default Value	Access Type	Description
Channel 1			
0	0	RW	Reserved
1	0	RW	RX8B10BEN[1]
2	0	RW	RXPOLARITY (0: Not-inverted 1: Inverted)
3	0	RW	RXPRBSCNTRESET

Table 38: **RX Control Register** *(cont'd)*

Bit	Default Value	Access Type	Description
7:4	0	RW	RXPRBSSEL[3:0] 7 series PRBS modes: 000 - Standard operation 001 - PRBS-7 010 - PRBS-15 011 - PRBS-23 100 - PRBS-31 UltraScale /UltraScale+ PRBS modes: 000 - Standard operation 001 - PRBS-7 010 - PRBS-9 010 - PRBS-15 011 - PRBS-23 100 - PRBS-31
Channel 2			
8	0	RW	Reserved
9	0	RW	RX8B10BEN[1]
10	0	RW	RXPOLARITY (0: Not-inverted 1: Inverted)
11	0	RW	RXPRBSCNTRESET
15:12	0	RW	RXPRBSSEL[3:0]. Refer to Channel 1 for PRBS modes.
Channel 3			
16	0	RW	Reserved
17	0	RW	RX8B10BEN[1]
18	0	RW	RXPOLARITY (0: Not-inverted 1: Inverted)
19	0	RW	RXPRBSCNTRESET
23:20	0	RW	RXPRBSSEL[3:0]. Refer to Channel 1 for PRBS modes.
Channel 4			
24	0	RW	Reserved
25	0	RW	RX8B10BEN[1]
26	0	RW	RXPOLARITY (0: Not-inverted 1: Inverted)
27	0	RW	RXPRBSCNTRESET
31:28	0	RW	RXPRBSSEL:0]. Refer to Channel 1 for PRBS modes.

Notes:

1. For the DisplayPort protocol, this field is tied to 1 for 7 series devices. For the HDMI protocol, this register field is unused.

RX Status (RXS) Register (0x0104)

Table 39: **RX Status (RXS) Register**

Bit	Default Value	Access Type	Description
Channel 1			
0	0	RO	RXCDRLOCK[1]
3:1	0	RO	RXBUFSTATUS [2:0]
4	0	RO	RXPRBSERR
7:5	0	RO	Reserved
Channel 2			
8	0	RO	RXCDRLOCK[1]
11:9	0	RO	RXBUFSTATUS [2:0]
12	0	RO	RXPRBSERR
15:13	0	RO	Reserved
Channel 3			
16	0	RO	RXCDRLOCK[1]
19:17	0	RO	RXBUFSTATUS [2:0]
20	0	RO	RXPRBSERR
23:21	0	RO	Reserved
Channel 4			
24	0	RO	RXCDRLOCK[1]
27:25	0	RO	RXBUFSTATUS [2:0]
28	0	RO	RXPRBSERR
31:29	0	RO	Reserved

Notes:

1. 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.

RX Equalization and CDR Register (0x0108)

Table 40: **RX Equalization and CDR Register**

Bit	Default Value	Access Type	Description
Channel 1			
0	0	RW	RXLPMEN
1	0	RW	RXCDRHOLD[1]
2	0	RW	RXOSOVRDEN[1]
3	0	RW	RXLPMLFKLOVRDEN[1]
4	0	RW	RXLPMHFOVRDEN[1]
5:7	0	RW	Reserved

Table 40: **RX Equalization and CDR Register** *(cont'd)*

Bit	Default Value	Access Type	Description
Channel 2			
8	0	RW	RXLPMEN
9	0	RW	RXCDRHOLD[1]
10	0	RW	RXOSOVRDEN[1]
11	0	RW	RXLPMLFKLOVRDEN[1]
12	0	RW	RXLPMHFOVRDEN[1]
15:13	0	RW	Reserved
Channel 3			
16	0	RW	RXLPMEN
17	0	RW	RXCDRHOLD[1]
18	0	RW	RXOSOVRDEN[1]
19	0	RW	RXLPMLFKLOVRDEN[1]
20	0	RW	RXLPMHFOVRDEN[1]
23:21	0	RW	Reserved
Channel 4			
24	0	RW	RXLPMEN
25	0	RW	RXCDRHOLD[1]
26	0	RW	RXOSOVRDEN[1]
27	0	RW	RXLPMLFKLOVRDEN[1]
28	0	RW	RXLPMHFOVRDEN[1]
31:29	0	RW	Reserved

Notes:

1. For the DisplayPort protocol, this register field is unused.

RX TDLOCK VALUE Register (0x010C)

For the DisplayPort protocol, this register field is unused.

Table 41: **RX TDLOCK VALUE Register**

Bit	Default Value	Access Type	Description
31:0	0	RW	[31:0]: TDLock value (Protocol Specific). Default value is 50UI.

RX Symbol Error Counter – Channel 1 and 2 Register (0x0084)

The error counter increments when there is a disparity error and 8B/10B symbol error. The counter is cleared after a read operation.

Table 42: **RX Symbol Error Counter – Channel 1 and 2 Register**

Bit	Default Value	Access Type	Description
15:0	0	RC[1]	Channel 1 error counter.
31:16	0	RC[1]	Channel 2 error counter.

Notes:

1. RC - Counter is cleared after read operation.

RX Symbol Error Counter – Channel 3 and 4 Register (0x0088)

Error counter increments when there is a disparity error and 8B/10B symbol error. The counter is cleared after a read operation.

Table 43: **RX Symbol Error Counter – Channel 3 and 4 Register**

Bit	Default Value	Access Type	Description
15:0	0	RC[1]	Channel 3 error counter.
31:16	0	RC[1]	Channel 4 error counter.

Notes:

1. RC - Counter is cleared after read operation.

Interrupts Register

Interrupt Enable Register (IER) (0x0110)

Table 44: **Interrupt Enable Register (IER)**

Bit	Default Value	Access Type	Description
0	0	WO	TX Reset Done Change Event
1	0	WO	RX Reset Done Change Event
2	0	WO	CPLL Lock Change Event
3	0	WO	QPLL Lock Change Event/PLL0 Lock Change Event (GTPE2)
4	0	WO	TX Alignment Done Event
5	0	WO	QPLL1 Lock Change Event/PLL1 Lock Change Event (GTPE2)
6	0	WO	Clock Detector TX Frequency Change
7	0	WO	Clock Detector RX Frequency Change
9	0	WO	TX MMCM Lock Change Event
10	0	WO	RX MMCM Lock Change Event
30	0	WO	Clock Detector TX Debounce Timeout

Table 44: **Interrupt Enable Register (IER)** *(cont'd)*

Bit	Default Value	Access Type	Description
31	0	WO	Clock Detector RX Debounce Timeout

Interrupt Disable Register (IDR) (0x0114)

Table 45: **Interrupt Disable Register (IDR)**

Bit	Default Value	Access Type	Description
0	0	WO	TX Reset Done Change Event
1	0	WO	RX Reset Done Change Event
2	0	WO	CPLL Lock Change Event
3	0	WO	QPLL/QPLL0 Lock Change Event/PLL0 Lock Change Event (GTPE2)
4	0	WO	TX Alignment Done Event
5	0	WO	QPLL1 Lock Change Event/PLL1 Lock Change Event (GTPE2)
6	0	WO	Clock Detector TX Frequency Change
7	0	WO	Clock Detector RX Frequency Change
9	0	WO	TX MMCM Lock Change Event
10	0	WO	RX MMCM Lock Change Event
30	0	WO	Clock Detector TX Debounce Timeout
31	0	WO	Clock Detector RX Debounce Timeout

Interrupt Mask Register (IMR) (0x0118)

Table 46: **Interrupt Mask Register (IMR)**

Bit	Default Value	Access Type	Description
0	0	WO	TX Reset Done Change Event
1	0	WO	RX Reset Done Change Event
2	0	WO	CPLL Lock Change Event
3	0	WO	QPLL/QPLL0 Lock Change Event/PLL0 Lock Change Event (GTPE2)
4	0	WO	TX Alignment Done Event
5	0	WO	QPLL1 Lock Change Event/PLL1 Lock Change Event (GTPE2)
6	0	WO	Clock Detector TX Frequency Change
7	0	WO	Clock Detector RX Frequency Change
9	0	WO	TX MMCM Lock Change Event
10	0	WO	RX MMCM Lock Change Event
30	0	WO	Clock Detector TX Debounce Timeout
31	0	WO	Clock Detector RX Debounce Timeout

Interrupt Status Register (ISR) (0x011C)

Valid interrupt event by AND of IMR and ISR.

Table 47: **Interrupt Status Register (ISR)**

Bit	Default Value	Access Type	Description
0	0	W1C	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	0	W1C	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	0	W1C	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	0	W1C	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	0	W1C	TX Alignment Done Event
5	0	W1C	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	0	W1C	Clock Detector TX Frequency Change
7	0	W1C	Clock Detector RX Frequency Change
9	0	W1C	TX MMCM Lock Change Event
10	0	W1C	RX MMCM Lock Change Event
30	0	W1C	Clock Detector TX Debounce Timeout
31	0	W1C	Clock Detector RX Debounce Timeout

TXUSRCLK Clocking Registers

MMCM TXUSRCLK Control/Status (MMCM_TXUSRCLK_CTRL) Register (0x0120)

Table 48: **MMCM TXUSRCLK Control/Status (MMCM_TXUSRCLK_CTRL) Register**

Bit	Default Value	Access Type	Description
0	0	RW	Reserved
1	0	RW	MMCM Reset
8	0	RW	Reserved
9	0	RO	MMCM Locked Status
10	0	RW	MMCM Power Down
11	0	RW	MMCM Lock Mask

DRP CONTROL MMCM TXUSRCLK Register (0x0124)

Table 49: **DRP CONTROL MMCM TXUSRCLK Register**

Bit	Default Value	Access Type	Description
11:0	0	RW	DRPADDR[8:0]
12	0	RW	DRPEN
13	0	RW	DRPWE
14	0	RW	DRP Reset (For UltraScale and UltraScale+ devices)[1]
15	0	RW	Reserved
31:16	0	RW	DRPDI [15:0]

Notes:

1. For the DisplayPort protocol, this register field is unused.

DRP STATUS MMCM TXUSRCLK Register (0x0128)

Table 50: **DRP STATUS MMCM TXUSRCLK Register**

Bit	Default Value	Access Type	Description
15:0	0	RO	DRPDO[15:0] – Valid for Read Transfers
16	0	RO	DRPRDY (Indicates valid transfer)
17	0	RO	DRPBUSY (Indicated DRP port free)
31:18	0	RO	Reserved

BUFGGT TXUSRCLK Control (BUFGGT_TXUSRCLK_CTRL) Register (0x0134)

BUFG_GT attributes register. For more details, see the *UltraScale Architecture Clocking Resources User Guide* (UG572).

Table 51: **BUFGGT TXUSRCLK Control (BUFGGT_TXUSRCLK_CTRL) Register**

Bit	Default Value	Access Type	Description
0	0	RW	CLR: Active-High asynchronous clear forcing BUFG_GT output to zero
3:1	0	RW	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

MISC TXUSRCLK Control (MISC_TXUSRCLK_CTRL) Register (0x0138)

Table 52: **MISC TXUSRCLK Control (MISC_TXUSRCLK_CTRL) Register**

Bit	Default Value	Access Type	Description
0	0	RW	CLKOUT1 OBUFTDS Output Enable. 0: disabled; 1: enabled HDMI TX TMDS Clock Output Enable
1	0	RW	TX REFCLK CEB
31:2	0	RW	Reserved

MMCM RXUSRCLK Control/Status (MMCM_RXUSRCLK_CTRL) Register (0x0140)

Table 53: **MMCM RXUSRCLK Control/Status (MMCM_RXUSRCLK_CTRL) Register**

Bit	Default Value	Access Type	Description
0	0	RW	Reserved
1	0	RW	MMCM Reset
8	0	RW	Reserved
9	0	RO	MMCM Locked Status
10	0	RW	MMCM Power Down
11	0	RW	MMCM Locked Mask
12	1	RW	MMCM Clock In Select 1 - CLK_IN1 0 - CLK_IN2

DRP CONTROL MMCM RXUSRCLK Register (0x0144)

Table 54: **DRP CONTROL MMCM RXUSRCLK Register**

Bit	Default Value	Access Type	Description
11:0	0	RW	DRPADDR[8:0]
12	0	RW	DRPEN
13	0	RW	DRPWE
14	0	RW	DRP Reset (For UltraScale and UltraScale+ devices)[1]
15	0	RW	Reserved
31:16	0	RW	DRPDI [15:0]

Notes:

1. For the DisplayPort protocol, this register field is unused.

DRP STATUS MMCM RXUSRCLK Register (0x0148)

Table 55: **DRP STATUS MMCM RXUSRCLK Register**

Bit	Default Value	Access Type	Description
15:0	0	RO	DRPDO[15:0] – Valid for Read Transfers
16	0	RO	DRPRDY (Indicates valid transfer)
17	0	RO	DRPBUSY (Indicated DRP port free)
31:18	0	RO	Reserved

BUFGGT RXUSRCLK Control (BUFGGT_RXUSRCLK_CTRL) Register (0x0154)

BUFG_GT attributes register. For more details, see the *UltraScale Architecture Clocking Resources User Guide* (UG572).

Table 56: **BUFGGT RXUSRCLK Control Register**

Bit	Default Value	Access Type	Description
0	0	RW	CLR: Active-High asynchronous clear forcing BUFG_GT output to zero
3:1	0	RW	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.

MISC RXUSRCLK Control (MISC_RXUSRCLK_CTRL) Register (0x0158)

Table 57: **MISC RXUSRCLK Control Register**

Bit	Default Value	Access Type	Description
0	0	RW	CLKOUT1 OBUFTDS Output Enables. 0 - disabled; 1 -enabled HDMI RX TMDS Clock Output Enable
1	0	RW	RX REFCLK CEB
31:2	0	RW	Reserved

Clock Detector (HDMI) Registers

Clock Detector (HDMI) Control Register (0x0200)

Table 58: **Clock Detector (HDMI) Control Register**

Bit	Default Value	Access Type	Description
0	0	RW	Run: Set this bit to enable clock detector.
1	0	RW	TX Timer Clear – Self Clearing
2	0	RW	RX Timer Clear – Self Clearing
3	0	RW	TX Frequency Reset – Self Clearing when TX Frequency Zero bit asserts
4	0	RW	RX Frequency Reset – Self Clearing when RX Frequency Zero bit asserts
12:5	0	RW	Frequency Lock Counter Threshold
16:13	8	RW	Clock detector accuracy range. A high value reduces the accuracy. Increase the value to get a stable operation when the incoming clock is not of good quality.

Clock Detector (HDMI) Status Register (0x0204)

Table 59: **Clock Detector (HDMI) Status Register**

Bit	Default Value	Access Type	Description
0	0	RO	TX Frequency Zero
1	0	RO	RX Frequency Zero
2	0	RO	TX Reference Clock Locked
3	0	RO	TX Reference Clock Locked Captured

Clock Detector (HDMI) Frequency Counter Timeout Register (0x0208)

Table 60: **Clock Detector (HDMI) Frequency Counter Timeout Register**

Bit	Default Value	Access Type	Description
31:0	0	RW	Clock Frequency in Hertz (Typically system frequency value)

Clock Detector (HDMI) Transmitter Frequency Register (0x020C)

Table 61: **Clock Detector (HDMI) Transmitter Frequency Register**

Bit	Default Value	Access Type	Description
31:0	0	RO	Transmitter Clock Frequency in Hertz

Clock Detector (HDMI) Receiver Frequency Register (0x0210)

Table 62: **Clock Detector (HDMI) Receiver Frequency Register**

Bit	Default Value	Access Type	Description
31:0	0	RO	Receiver Clock Frequency in Hertz

Clock Detector (HDMI) Transmitter Timer Register (0x0214)

Table 63: **Clock Detector (HDMI) Transmitter Timer Register**

Bit	Default Value	Access Type	Description
31:0	0	RW	Transmitter Timeout Value at System Clock per tick

Clock Detector (HDMI) Receiver Timer Register (0x0218)

Table 64: **Clock Detector (HDMI) Receiver Timer Register**

Bit	Default Value	Access Type	Description
31:0	0	RW	Receiver Timeout Value at System Clock per tick

Clock Detector (HDMI) DRU Frequency Register (0x021C)

Table 65: **Clock Detector (HDMI) DRU Frequency Register**

Bit	Default Value		Description
31:0	0	RO	Data Recovery Unit Clock Frequency in Hertz

Data Recovery Unit Registers

Data Recovery Unit Control Register (0x0300)

Table 66: **Data Recovery Unit Control Register - All Channels**

Bit	Default Value	Access Type	Description
0	0	RW	Reset - 1 – Reset asserted; 0 – Reset released
1	0	RW	Enable - 0 – DRU disabled; 1 – DRU enabled
31:2	0	RW	Reserved

Data Recovery Unit Status Register (0x0304)

Table 67: **Data Recovery Unit Status Register**

Bit	Default Value	Access Type	Description
0	0	RO	Channel 1 DRU Active
1	0	RO	Channel 2 DRU Active
2	0	RO	Channel 3 DRU Active
3	0	RO	Channel 4 DRU Active
23:4	0	RO	Reserved
31:24	0	RO	DRU Version

Center Frequency Low Register – All Channels Register (0x0308)

Table 68: **Center Frequency Low Register – All Channels Register**

Bit	Default Value	Access Type	Description
31:0	0	RW	Center frequency bits 31:0

Center Frequency High Register – All Channels Register (0x030C)

Table 69: **Center Frequency High Register – All Channels Register**

Bit	Default Value	Access Type	Description
4:0	0	RW	Center frequency bits 36:32
31:5	0	RW	Reserved

Gain Register – All Channels (0x0310)

Table 70: **Gain Register – All Channels**

Bit	Default Value	Access Type	Description
4:0	0	RO	Gain G1
7:5	0	RO	Reserved
12:8	0	RO	Gain G1 P
15:13	0	RO	Reserved
20:16	0	RO	Gain G2
31:21	0	RO	Reserved

TX TMDS Pattern Generator Control Register (0x0340)

Table 71: **TX TMDS Pattern Generator Control Register**

Bit	Default Value	Access Type	Description
2:0	0	RW	Clock Ratio 0x0 - OFF 0x1 - Ratio - 10 0x2 - Ratio - 20 0x3 - Ratio - 30 0x4 - Ratio - 40 0x5 - Ratio - 50 0x6 - OFF 0x7 - OFF

Designing with the Core

This section 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. See the following for more information.

- *7 Series FPGAs GTX/GTH Transceivers User Guide* (UG476)
- *UltraScale Architecture GTH Transceivers User Guide* (UG576)
- *7 Series FPGAs GTP Transceivers User Guide* (UG482)
- *UltraScale Architecture GTY Transceivers User Guide* (UG578)

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

The frequency details of the reference clock and the link clock are provided in the following table.

Table 72: **Reference Clock and Link Clock Frequency Details of DisplayPort 1.2 TX**

Family	Link Rate (Gb/s)	TX Link Clock (MHz)		Reference Clock Source	Reference Clock Frequency (MHz)	PLL Type
		GT Data Width = 2	GT Data Width = 4			
Kintex-7 GTX	1.62	81	40.5	On-board clock	135	CPLL
	2.7	135	67.5	On-board clock	135	CPLL
	5.4	270	135	On-board clock	135	CPLL
Virtex-7 GTH	1.62	81	40.5	On-board clock	135	CPLL
	2.7	135	67.5	On-board clock	135	CPLL
	5.4	270	135	On-board clock	135	CPLL

Table 72: **Reference Clock and Link Clock Frequency Details of DisplayPort 1.2 TX** *(cont'd)*

Family	Link Rate (Gb/s)	TX Link Clock (MHz)		Reference Clock Source	Reference Clock Frequency (MHz)	PLL Type
		GT Data Width = 2	GT Data Width = 4			
UltraScale™ GTH3 UltraScale+™ GTHE4	1.62	81	40.5	On-board clock	135	QPLL1
	2.7	135	67.5	On-board clock	135	QPLL1
	5.4	270	135	On-board clock	135	QPLL1

Table 73: **Reference Clock and Link Clock Frequency Details of DisplayPort 1.2 RX**

Family	Link Rate (Gb/s)	RX Link Clock (MHz)		Reference Clock Source	Reference Clock Frequency (MHz)	PLL Type
		GT Data Width = 2	GT Data Width = 4			
Kintex-7 GTX	1.62	81	40.5	DP159	81	CPLL
	2.7	135	67.5	DP159	135	CPLL
	5.4	270	135	DP159	270	CPLL
Virtex-7 GTH	1.62	81	40.5	DP159	81	CPLL
	2.7	135	67.5	DP159	135	CPLL
	5.4	270	135	DP159	270	CPLL
UltraScale GTH3 UltraScale+ GTHE4	1.62	81	40.5	DP159	81	CPLL
	2.7	135	67.5	DP159	135	CPLL
	5.4	270	135	DP159	270	CPLL

Table 74: **Reference Clock and Link Clock Frequency Details of DisplayPort 1.4 TX**

Family	Link Rate (Gb/s)	TX Link Clock (MHz)	Reference Clock Source	Reference Clock Frequency (MHz)	PLL Type
		GT Data Width = 2			
UltraScale GTH3 UltraScale+ GTHE4/GTYE4	1.62	81	On-board clock	270	QPLL1
	2.7	135	On-board clock	270	QPLL1
	5.4	270	On-board clock	270	QPLL1
	8.1	405	On-board clock	270	QPLL1

Table 75: **Reference Clock and Link Clock Frequency Details of DisplayPort 1.4 RX**

Family	Link Rate (Gb/s)	RX Link Clock (MHz)	Reference Clock Source	Reference Clock Frequency (MHz)	PLL Type
		GT Data Width = 2			
UltraScale GTH3 UltraScale+ GTHE4/GTYE4	1.62	81	On-board clock	270	CPLL
	2.7	135	On-board clock	270	CPLL
	5.4	270	On-board clock	270	CPLL
	8.1	405	On-board clock	270	CPLL

IMPORTANT! *Transmit Buffer Bypass must be enabled for the DisplayPort version 1.2 specification PHY compliance. This is required to pass the Inter-pair Skew Test in the DisplayPort 1.2 specification source device compliance tests; this test is only informative for the DisplayPort 1.4 specification compliant source.*

IMPORTANT! *Digital VCXO Replacement (PICXO) is not supported by the Video PHY Controller. For a field-customized Video PHY Controller that use PICXO, the Transmit Buffer Bypass must be disabled, as it is a PICXO requirement. The Video PHY Controller is started up with Transmit Buffer Bypass enabled, and then switched to Transmit Buffer Bypass disabled during PICXO operation.*

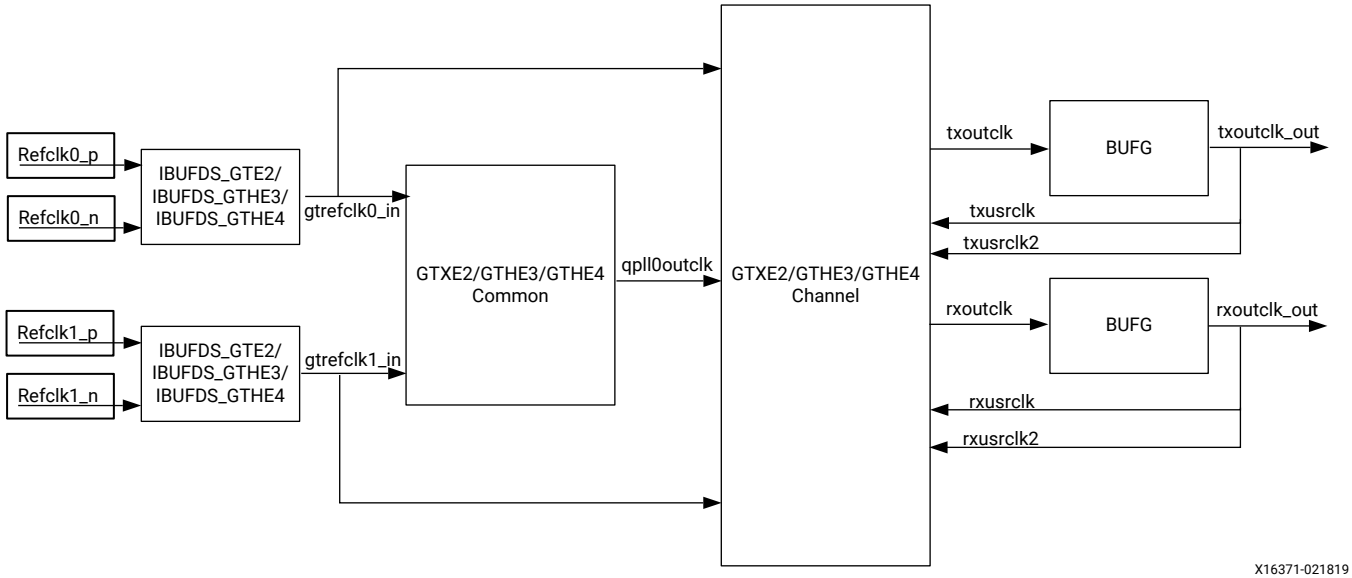
GTXE2*/GTHE3 Clocking When Transmit Buffer Bypass is Disabled

For DisplayPort 1.2, connect the DP159 forwarded clock to any of the receive reference clock input. An external 135 MHz oscillator is required for transmit (this clock cannot be forwarded from the fabric). Use a valid reference clock selection through the driver API.

For DisplayPort 1.4, connect an external oscillator generated 270 MHz clock to the reference clock inputs of the transmitter and receiver (this clock cannot be forwarded from the fabric). Use a valid reference clock selection through the driver API. In DisplayPort 1.4, the Retimer forwarded clock is not used.

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

Figure 4: **GTXE2/GTHE3 Clocking (TX Buffer Bypass = Disabled)**



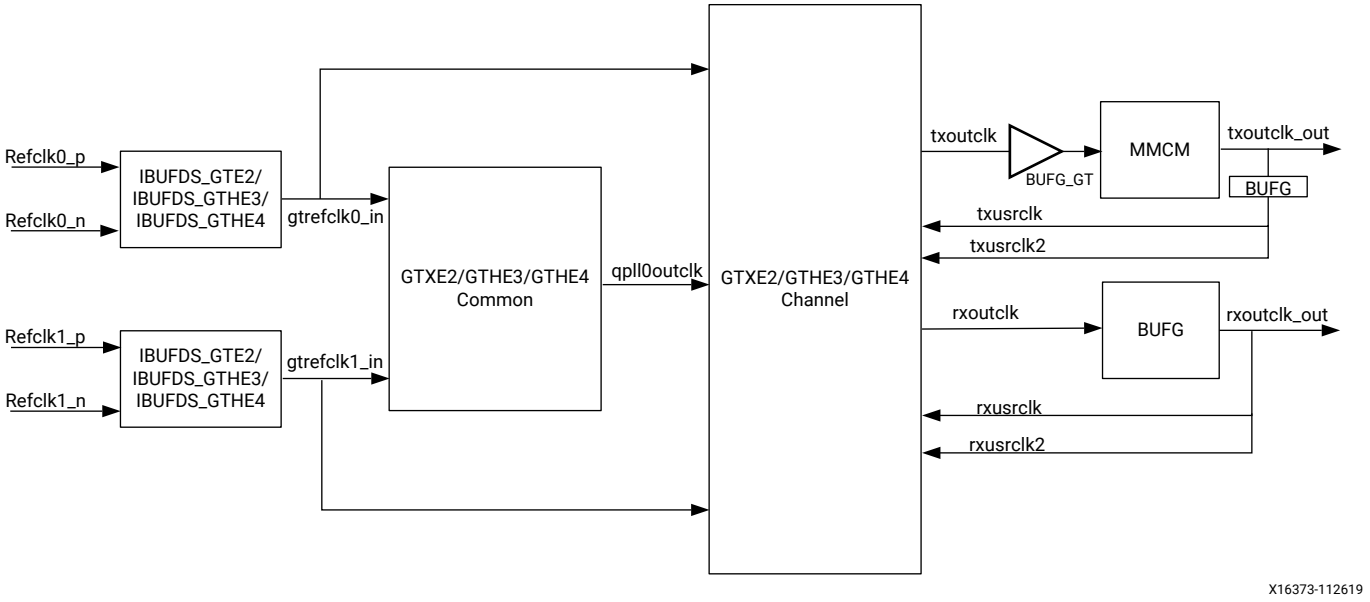
GTXE2*/GTHE3 Clocking When Transmit Buffer Bypass is Enabled

For DisplayPort 1.2, connect the DP159 forwarded clock to any of the reference clock inputs and use a valid reference clock selection through the driver API.

For DisplayPort 1.4, connect a 270 MHz clock to any of the reference clock inputs and use a valid reference clock selection using the driver API. In DisplayPort 1.4, the Retimer forwarded clock is not used.

The `txoutclk_out/rxoutclk_out` signals are connected to the DisplayPort MAC controller. In TX buffer bypass mode, the MMCM clocking resource is used on the TX path and you have to program valid divider/multiplier values using the driver API.

Figure 5: **GTXE2/GTHE3/GTHE4 Clocking (TX Buffer Bypass = Enabled)**



***Note*:** GTXE2 is supported only in DisplayPort 1.2 subsystems.

The system clock must drive the `vid_phy_sb_aclk`, `vid_phy_axi4lite_aclk`, and `drpclk` (for UltraScale+/UltraScale devices only). The ports should be connected to a valid clock, for example, 50 MHz, 100 MHz, or 150 MHz. The system clock must have sufficient buffer, for example, BUFG, before it can be used and connected.

HDMI Clocking

IMPORTANT! *The Transmit Buffer Bypass is always enabled for HDMI PHY compliance.*

The HDMI core clocking diagrams per transceiver type are shown below. Follow the guidelines below when connecting the Video PHY Controller 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 Video PHY Controller Vivado® IDE. The TX reference clock lock indicator should be connected to the `tx_refclk_rdy` port. See HDMI Reference Clock Requirements 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 Video PHY Controller IDE.
- If NI-DRU is enabled, connect the DRU reference clock to the DRU reference clock input that was selected in the Video PHY Controller IDE. See HDMI Reference Clock Requirement for the NI-DRU frequency requirements.
- The `txoutclk_out/rxoutclk_out` signal is connected to the `link_clk` of the HDMI MAC controller.
- The `tx_video_out/rx_video_out` signals are connected to the `video_clk` of 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, for example, audio generator module.
- The `rx_tmds_clk_p/n` signal should be connected to the input of the external clock generator if the Video PHY Controller is used in pass through mode. This is to have a phase aligned and jitter attenuated reference clock for the HDMI TX Subsystem.
- The `rx_tmds_clk` signal can be connected to any logic.

***Note*:** The HDMI RX and TX Subsystem clocks must be phase aligned in pass through mode to ensure seamless video streaming. Otherwise, the video output intermittently breaks due to mismatching clocks. This connection is not needed if the Video PHY Controller is used in a TX-only application because the external clock generator should run in standalone mode, using its local oscillator as its reference.

The following clocking diagrams show the default clock buffers used per device type. You can change these buffers as per your application's requirements using the user configurable parameters. These buffers can be changed by users according to their own application, through the user configurable parameters. These parameters are in white dash-lined boxes with prefix `CONFIG.<user_param_name>`.

IMPORTANT! *The Video PHY Controller has been tested using the default settings. You are expected to understand the proper clock buffer usage and design implications when changing the user parameters. See the 7 Series FPGAs Clocking Resources User Guide (UG472) and the UltraScale Architecture Clocking Resources User Guide (UG572).*

The user parameters can be configured using Tcl commands or through the Block Properties window in IP integrator. For example:

```
set_property -dict [list CONFIG.C_Use_Oddr_for_Tmds_Clkout {false}]
[get_ips <ip name>]
set_property -dict [list CONFIG.C_Tx_Outclk_Buffer {none}] [get_ips <ip
name>]
set_property -dict [list CONFIG.C_Rx_Video_Clk_Buffer {bufg}] [get_ips <ip
name>]
```

Related Information

HDMI Video PHY Controller Example Design
HDMI Reference Clock Requirements

GTXE2

Figure 6: **GTXE2**

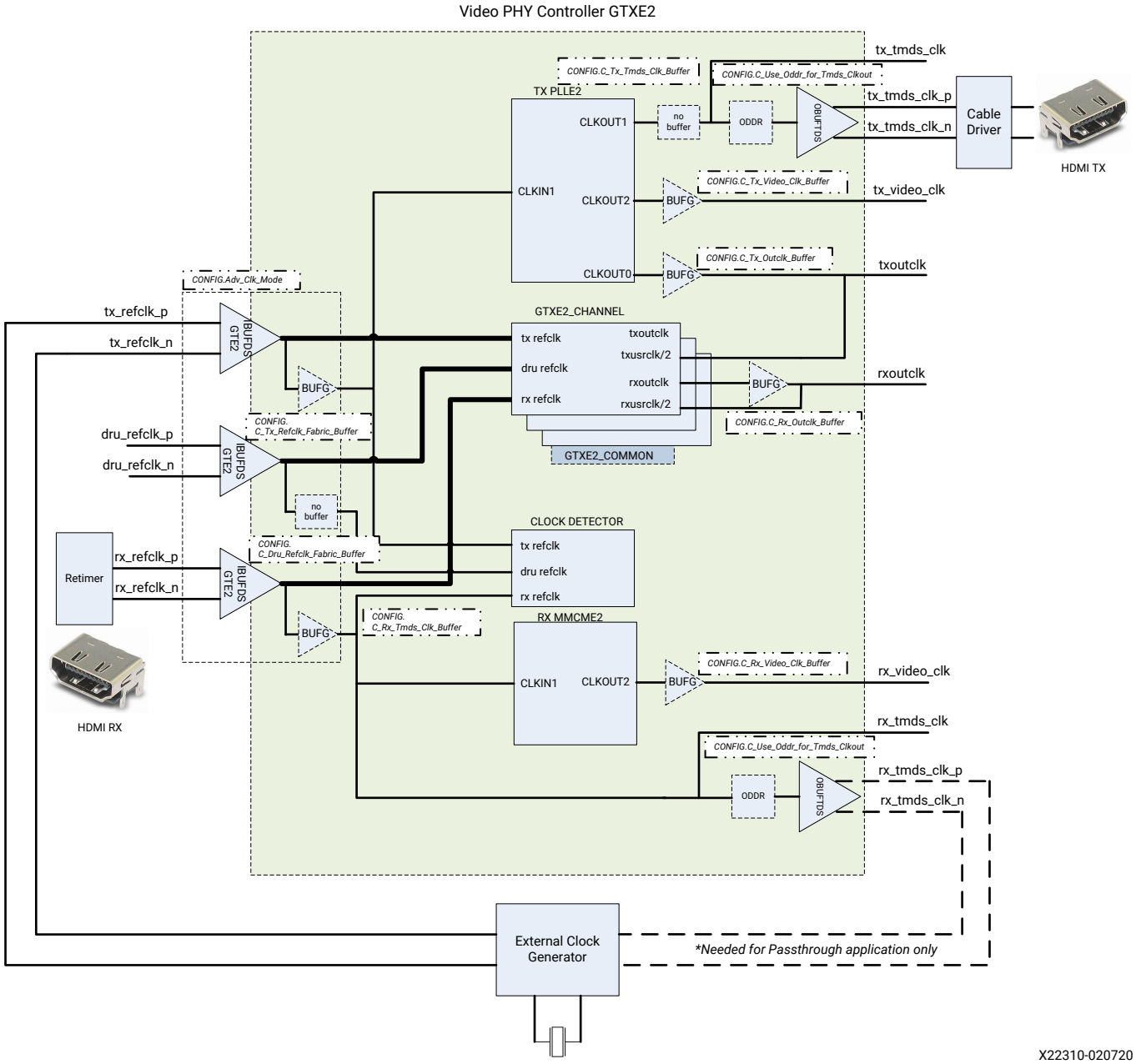
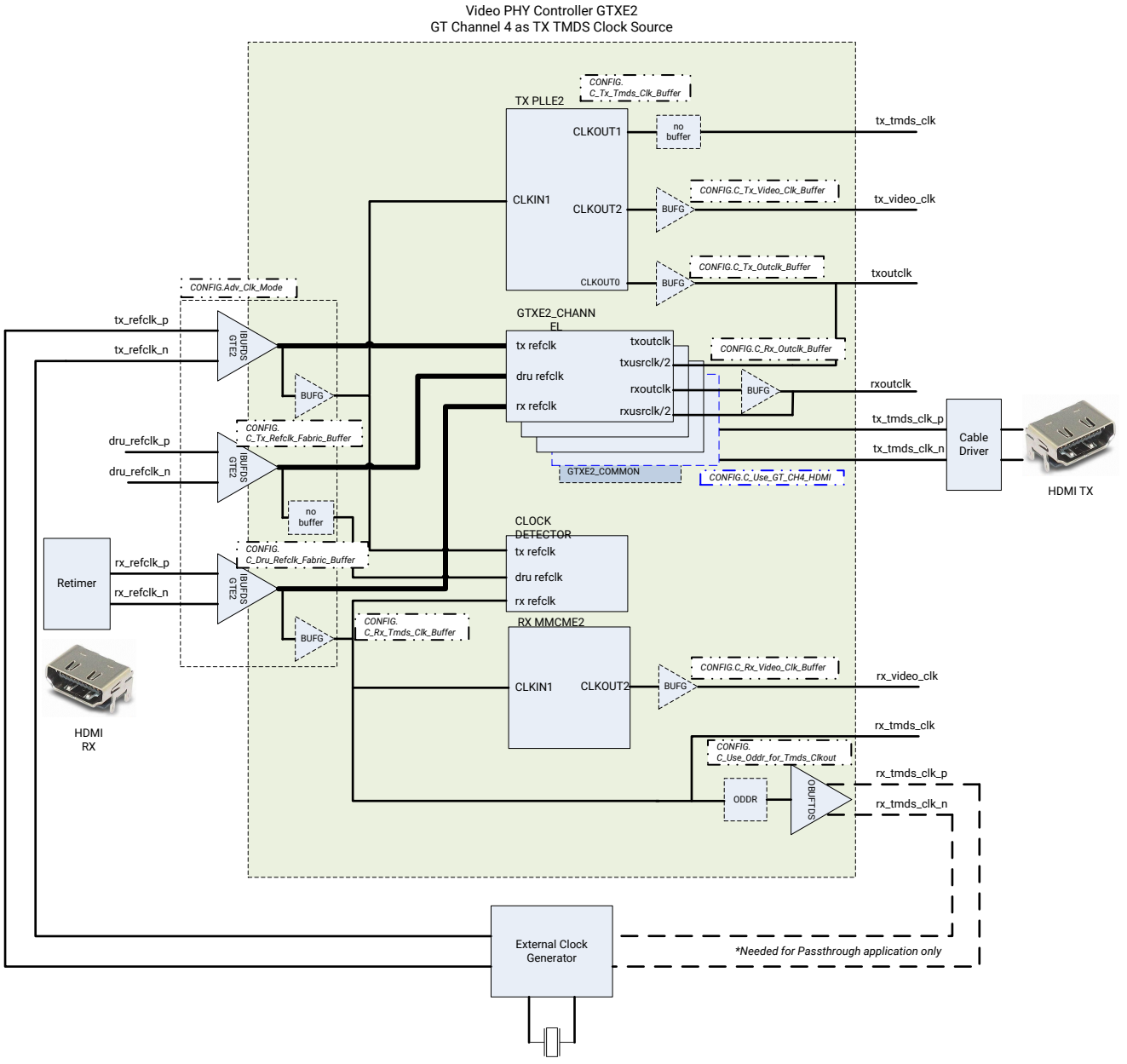


Figure 7: **Video PHY Controller GTXE2 GT Channel 4 as TX TMDS Clock Source**



The following parameters affect the clocking of GTXE2 device.

- **Adv_Clk_Mode:** Configured through a check box in GUI.

  This controls where the IBUFDS_GTE2 clock buffers of MGTREFCLK0 and MGTREFCLK1 are placed. When disabled, IBUFDS_GTE2 is placed within VPHY, but when enabled, IBUFDS_GTE2 should be manually instantiated at the system level. This is an ideal mode for applications requiring reference clock sharing across multiple VPHY instances.

- **C_Tx_Refclk_Fabric_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the TX REFCLK input clock.

  Valid parameters: none, bufg, bufh, bufmr, and bufr.

- **C_Dru_Refclk_Fabric_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the DRU REFCLK input clock.

  Valid parameters: none, bufg, bufh, bufmr, and bufr.

- **C_Rx_Tmds_Clk_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the RX REFCLK input clock.

  Valid parameters: none, bufg, bufh, bufmr, and bufr.

- **C_Tx_Tmds_Clk_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the TX TMDS output clock

  Valid parameters: none, bufg, bufh, bufmr, and bufr.

- **C_Tx_Video_Clk_Buffer :** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the TX Video output clock.

  Valid parameters: none, bufg, bufh, bufmr, and bufr.

- **C_Tx_Outclk_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the TX Link output clock.

  Valid parameters: none, bufg, bufh, bufmr, and bufr.

- **C_Rx_Outclk_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the RX Link output clock.

  Valid parameters: none, bufg, bufh, bufmr, and bufr.

- **C_Rx_Video_Clk_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the RX Video output clock.

  Valid parameters: none, bufg, bufh, bufmr, and bufr.

- **C_Use_Oddr_for_Tmds_Clkout:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls whether an ODDR is inserted to drive the OBUFTDS for differential TX and RX TMDS output clocks.

Valid parameters: TRUE or FALSE.

- **C_Use_GT_CH4_HDMI:** Configured through a check box in the Vivao IDE, through a Tcl command, or through the Block Properties Window in IP integrator. This controls whether a fourth GT channel is used in generating and transmitting the differential TX TMDS Clock (`tx_tmds_clk_p/n`).

  Valid parameters: TRUE or FALSE.

GTPE2

Figure 8: **GTPE2**

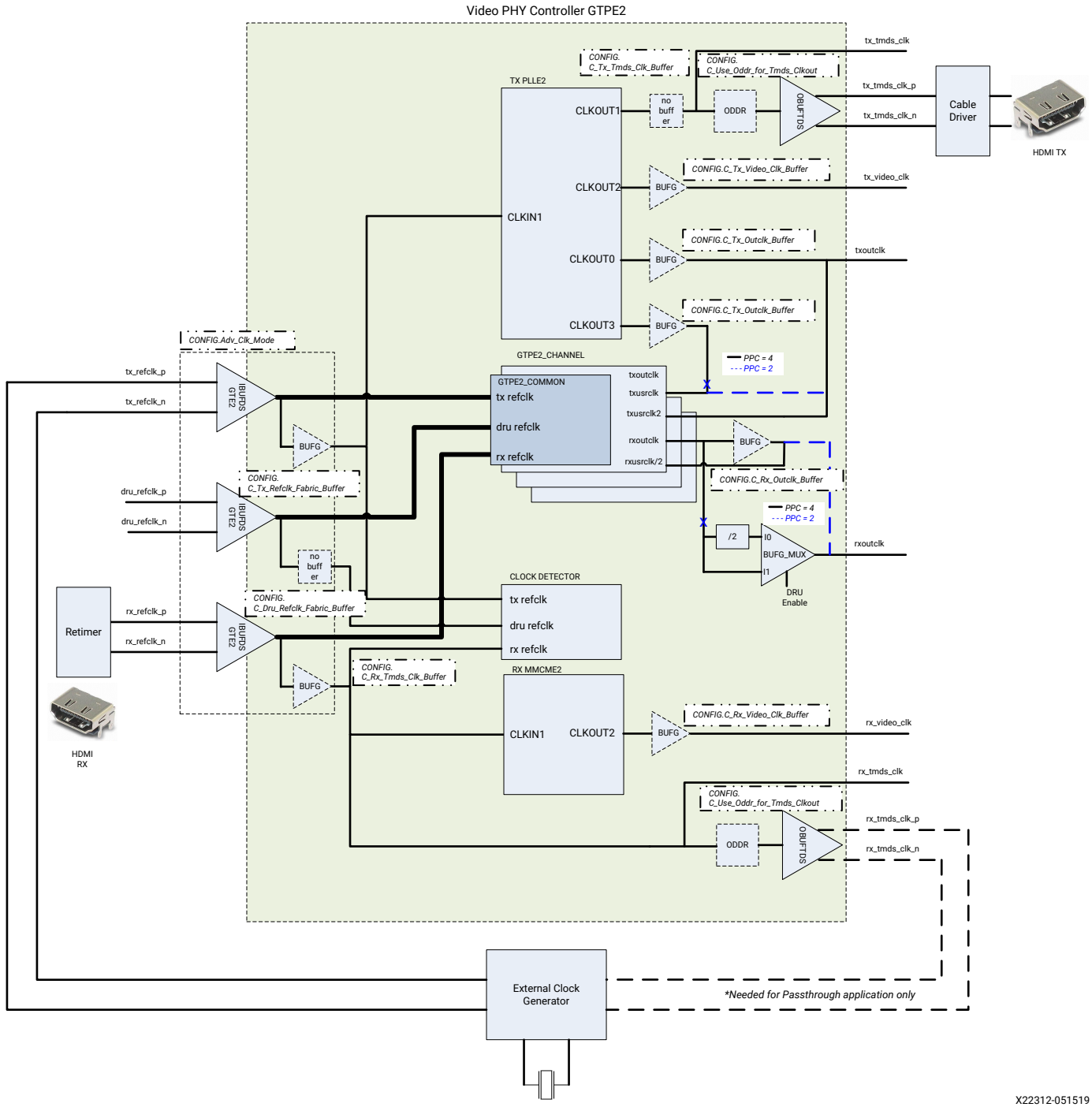
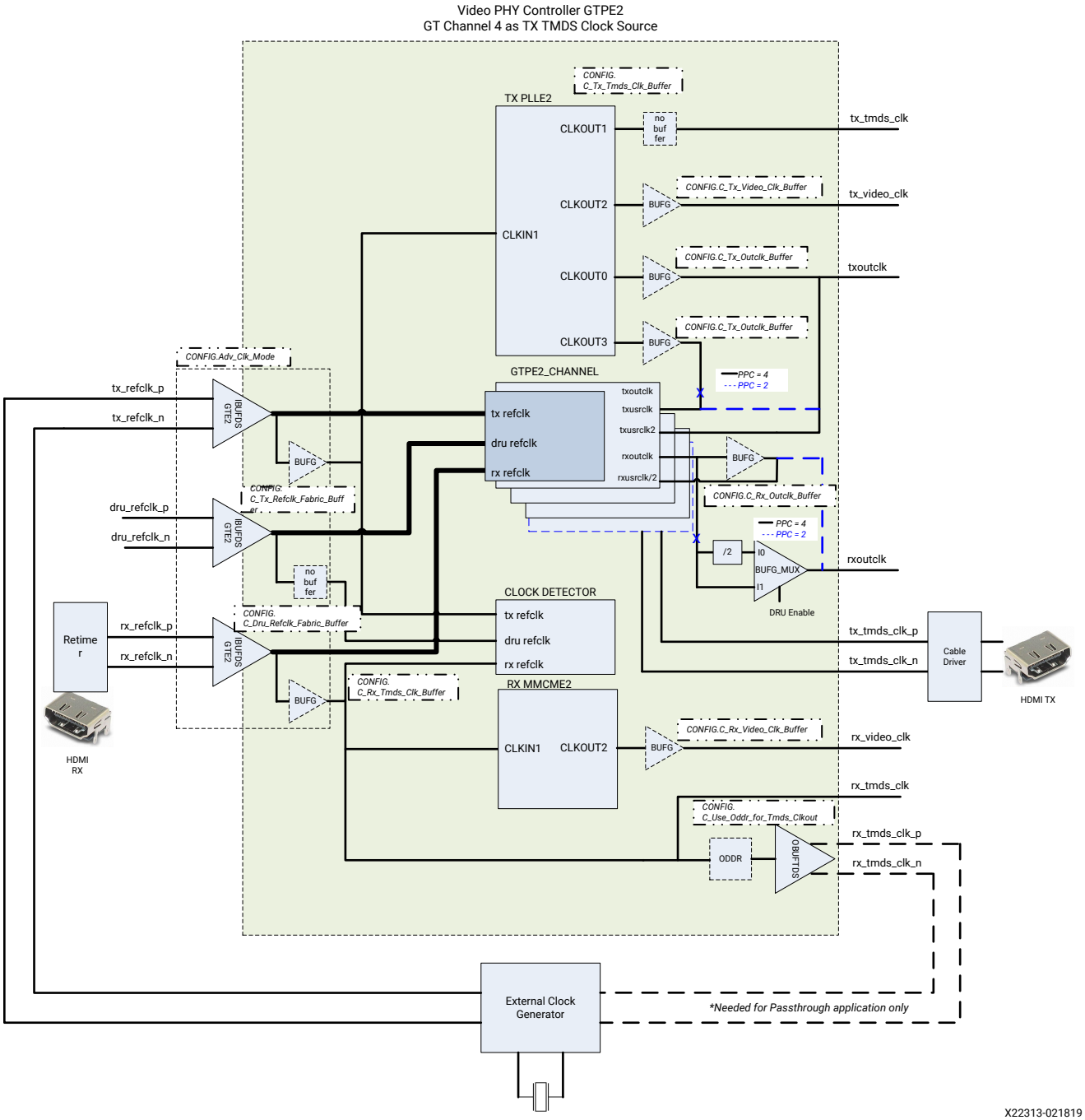


Figure 9: **Video PHY Controller GTPE2 GT Channel 4 as TX TMDS Clock Source**



The following parameters affect the clocking of GTPE2 device.

- **Adv_Clk_Mode:** Configured through a check box in the Vivado IDE.

This controls where the IBUFDS_GTE2 clock buffers of MGTREFCLK0 and MGTREFCLK1 are placed. When disabled, IBUFDS_GTE2 is placed within VPHY, but when enabled, IBUFDS_GTE2 should be manually instantiated at the system level. This is an ideal mode for applications requiring reference clock sharing across multiple VPHY instances.

- **C_Tx_Refclk_Fabric_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the TX REFCLK input clock.

  Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Dru_Refclk_Fabric_Buffer:** Configured through Tcl command or through the Block Properties Window in IP Integrator. This controls the type of buffer to be used for driving the fabric associated with the DRU REFCLK input clock.

  Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Rx_Tmds_Clk_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the RX REFCLK input clock.

  Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Tx_Tmds_Clk_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the TX TMDS output clock

  Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Tx_Video_Clk_Buffer :** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the TX Video output clock.

  Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Tx_Outclk_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the TX Link output clock.

  Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Rx_Outclk_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the RX Link output clock.

  Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Rx_Video_Clk_Buffer:** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the RX Video output clock.

Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Use_Oddr_for_Tmds_Clkout :** Configured through Tcl command or through the Block Properties Window in IP integrator. This controls whether an ODDR is inserted to drive the OBUFTDS for differential TX and RX TMDS output clocks.

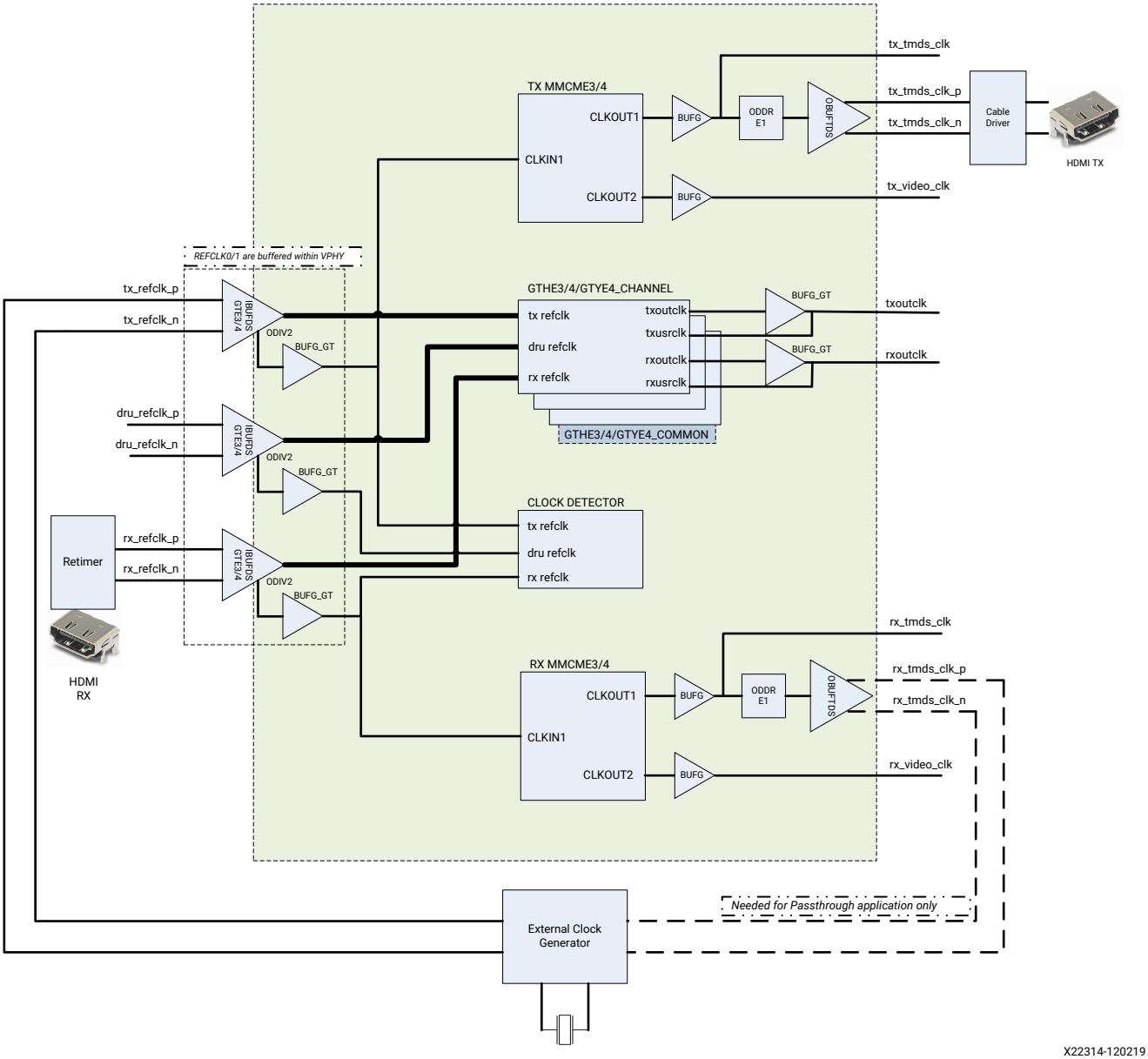
  Valid parameters are: TRUE or FALSE.

- **C_Use_GT_CH4_HDMI:** Configured through a check box in the Vivado IDE, through a Tcl command, or through the Block Properties Window in IP integrator. This controls whether a fourth GT channel is used in generating and transmitting the differential TX TMDS clock (`tx_tmds_clk_p/n`).

  Valid parameters are: TRUE or FALSE.

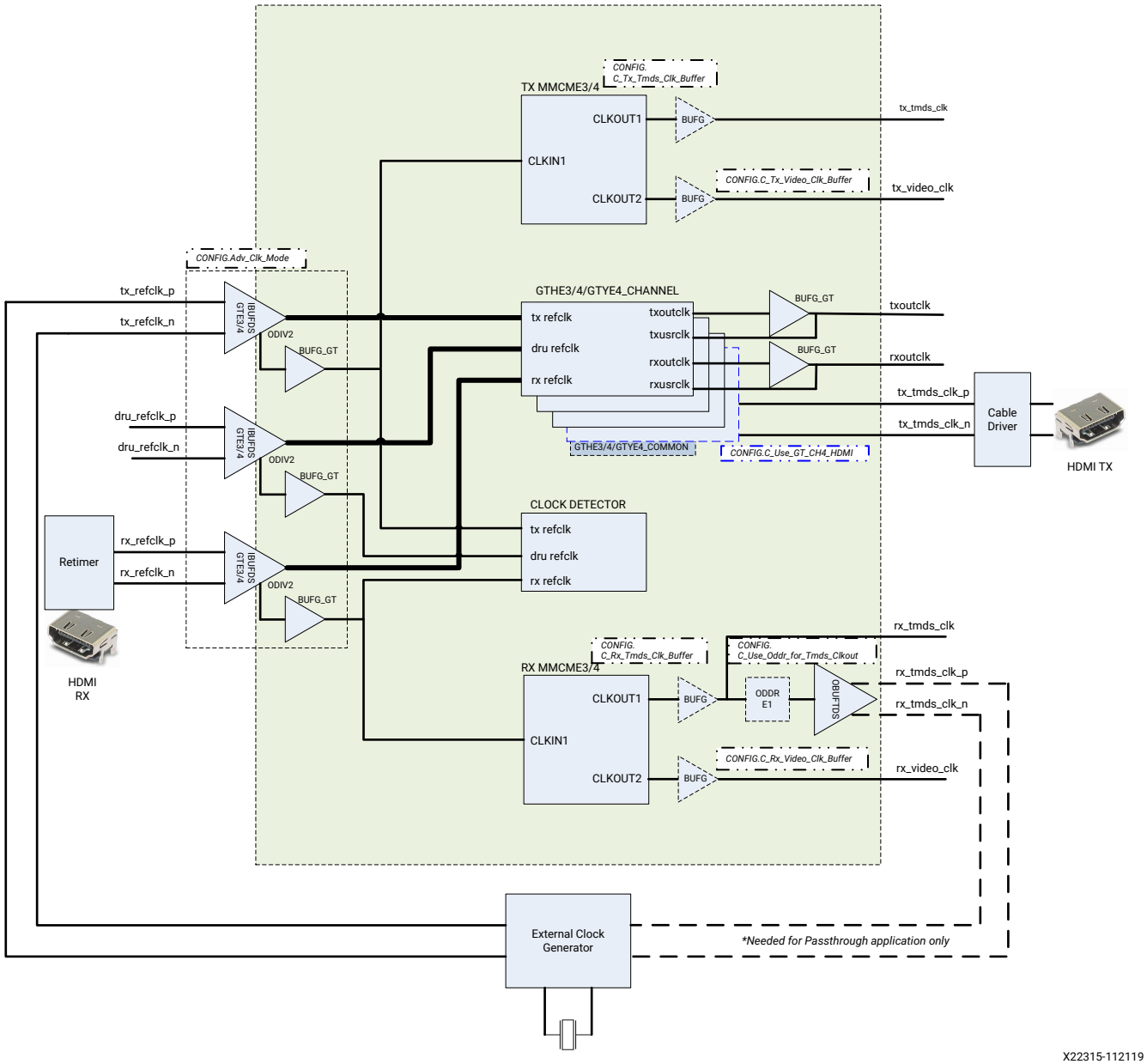
GTHE3, GTHE4, and GTYE4

Figure 10: **GTHE3, GTHE4, and GTYE4**



X22314-120219

Figure 11: **Video PHY Controller GTHE3, GTHE4 & GTYE4**



The following parameters affect the clocking of GTHE3, GTHE4, and GTYE4 devices.

- **Adv_Clk_Mode:** Configured through a check box in the Vivado IDE.

  This controls where the IBUFDS_GTE3/4 clock buffers of MGTREFCLK0 and MGTREFCLK1 are placed. When disabled, IBUFDS_GTE3/4 is placed within the Video PHY Controller, but when enabled, IBUFDS_GTE2 should be manually instantiated at the system level. This is an ideal mode for applications requiring reference clock sharing across multiple Video PHY Controller instances.

- **C_Rx_Tmds_Clk_Buffer:** Configured through a Tcl command or through the Block Properties window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with the RX TDMS output clock.

  Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Tx_Tmds_Clk_Buffer:** Configured through a Tcl command or through the Block Properties window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with theTX TMDS output clock

  Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Tx_Video_Clk_Buffer :** Configured through a Tcl command or through the Block Properties window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with theTX Video output clock.

  Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Rx_Video_Clk_Buffer:** Configured through a Tcl command or through the Block Properties window in IP integrator. This controls the type of buffer to be used for driving the fabric associated with theRX Video output clock.

  Valid parameters are: none, bufg, bufh, bufmr, and bufr.

- **C_Use_Oddr_for_Tmds_Clkout :** Configured through a Tcl command or through the Block Properties window in IP integrator. This controls whether an ODDRE1 is inserted to drive the OBUFTDS for differential TX and RX TMDS output clocks.

  Valid parameters are: TRUE or FALSE.

***Note*:** The BUFG_GTs that are directly connected to the transceiver are a hard requirement for clocks that are either driving or being driven by the GTHE3, GTHE4, and GTYE4 transceivers. They cannot be disabled or removed from the RTL.

Connecting an UltraScale/UltraScale+ GT NORTH Reference Clock to a CPLL

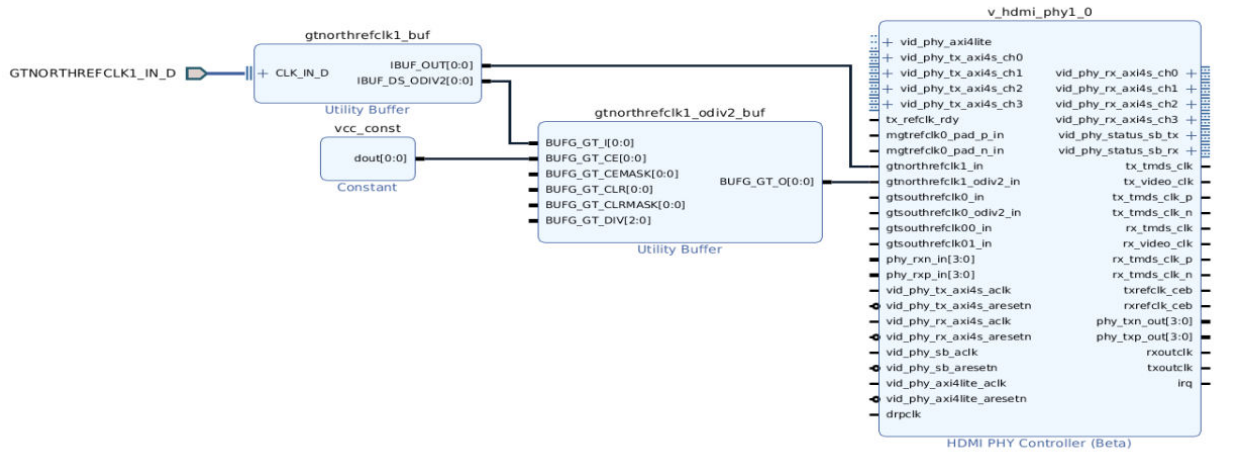
The Video PHY Controller opens the North and South reference clocks ports differently, depending on which PLL they are associated with. Additional ports with suffix 00 and 01 are opened when a reference clock is used with QPLL0/1. See Connecting an UltraScale/UltraScale+ GT South Reference Clock to a 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 the following figure.

- `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 the following figure.

Figure 12: **North Reference Clock**



***Note*:** BUFG_GT_CE port of the BUFG_GT buffer must be driven High.

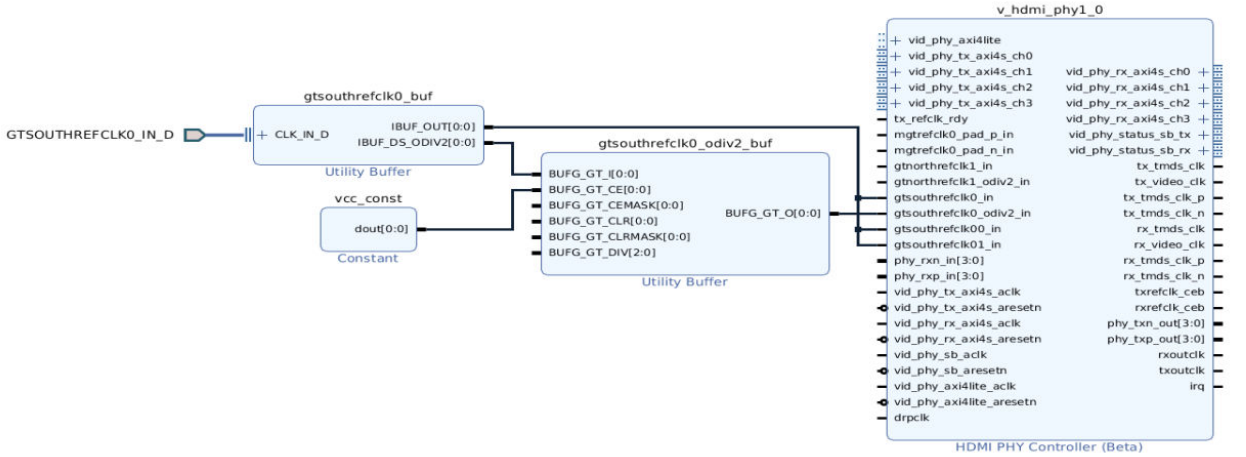
***Note*:** IBUF_DS_ODIV2 port of the IBUFDSGTE is by default configured to output divide by 1.

Connecting an UltraScale/UltraScale+ GT South Reference Clock to a QPLL0/1

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, the 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 the following figure.
- `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 the following figure.
- `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 the following figure.
- `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 the following figure.

Figure 13: **South Reference Clock**



X22693-051319

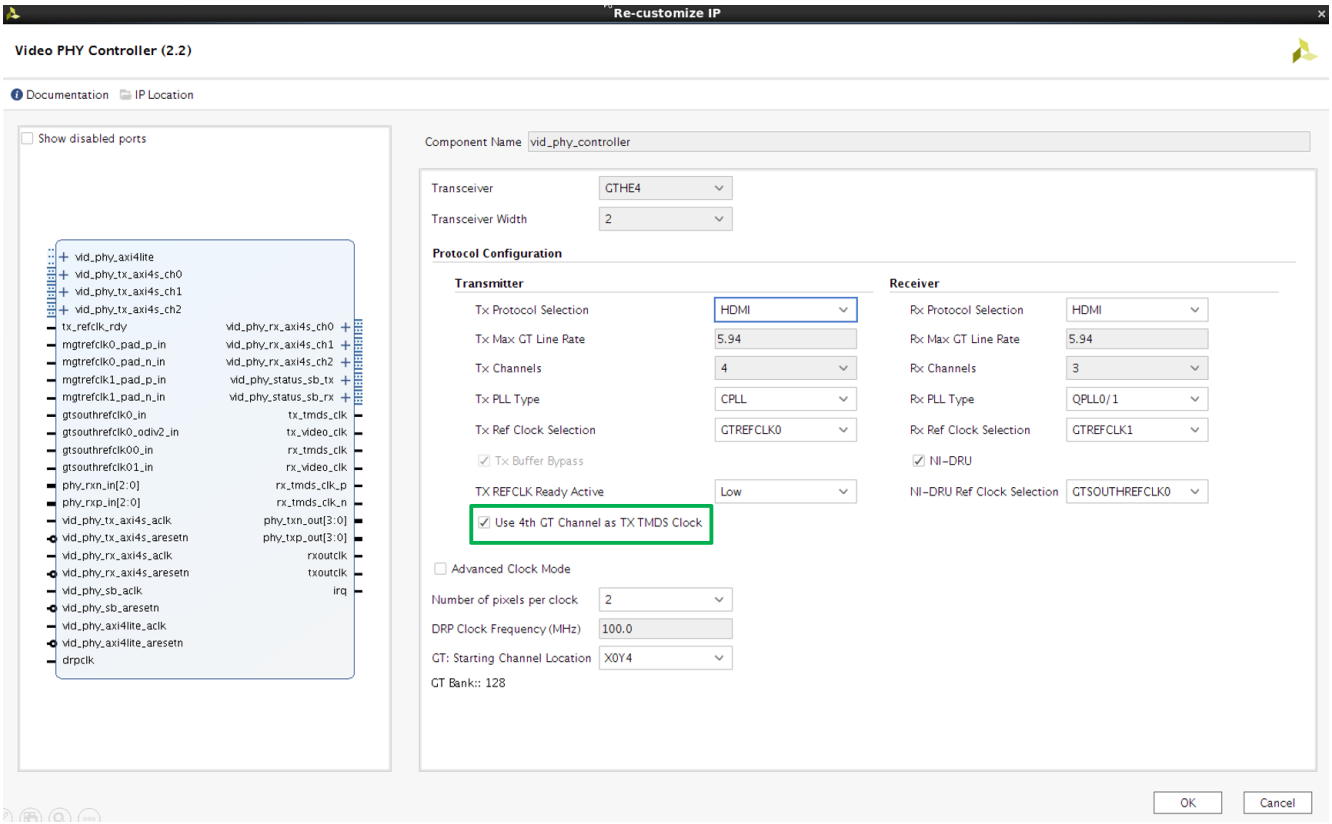
***Note*:** BUFG_GT_CE port of the BUFG_GT buffer must be driven High.

***Note*:** IBUF_DS_ODIV2 port of the IBUFDSGTE is by default configured to output divide by 1.

Using Fourth GT Channel as TX TMDS Clock Source

HDMI 1.4 and 2.0 protocols use three GT channels in a Quad leaving one unused. The fourth GT channel can be enabled and used as the TX TMDS Clock source instead of being generated by DCM (MMCM or PLL) This is done using a TCL command, or through the block properties window in IP integrator.

Figure 14: **4th GT Channel as TX TMDS Clock**



When this option is enabled, the TX TMDS Clock ports, `tx_tmds_clk_p/n` output ports are disabled and are replaced by `phy_txn/p_out[3]`.

A pattern generator module is added in the Video PHY Controller architecture to generate the specific pattern needed to generate the required TMDS clock frequency from the 4th GT channel. The pattern generator control register is located at offset 0x340 and is programmed based on the line rate to TMDS clock ratio. The pattern generator supports ratio of 10, 20, 30, 40, and 50. For example, in a typical HDMI 1.4 resolution such as 1080p60, the line rate per channel is 1.485 Gb/s and the TMDS clock is 148.5 MHz, thus giving a ratio of 10. For low line rate resolutions such as 480P60 which needs an oversampling technique (e.g. x3) to be transmitted, the ratio is computed as actual line rate per channel (270 Mb/s x 3) over TMDS clock (27 MHz), which gives a ratio of 30. For HDMI 2.0 resolution such as 4KP60, the line rate is 5.94 Gb/s and TMDS clock is 148.5 MHz, thus giving a ratio of 40.

Take note that a corresponding pin assignment constraint must be added in the top-level XDC for designs targeting 7 series devices such as Artix-7 and Kintex-7, to properly connect the `phy_txn/p_out[3]` ports to its corresponding MGT pins. The additional constraint is not needed when targeting UltraScale and UltraScale+ devices because the pin assignments are already handled within the GT Wizard in the Video PHY Controller. If the pin assignment constraints in the top-level XDC are different, the pin assignment generated within the GT wizard overwrites the ones in the top-level XDC and might cause critical warning messages.

Additional device LUTs and flip-flops are consumed when the 4th GT Channel is enabled+- as the TX TMDS clock source. Be aware of these additional resources when enabling this option.

RECOMMENDED: *For designs with high FPGA utilization or PCB noise, use the 4th GT channel for better signal integrity.*

Table 76: **HDMI 2.1 FRL Max Line Rates Supported Across Devices**

MAX FRL Rate		12Gbps @ 4 Lanes	10Gbps @ 4 Lanes	8Gbps @ 4 Lanes	6Gbps @ 4 Lanes	6Gbps @ 3 Lanes	3Gbps @ 3 Lanes	TMDS
Device Family	Speed Grade							
Zynq UltraScale +	-1LV	Not Supported		Supported				
	-1, -1L, -2LV	Not Supported	Supported					
	-2, -2L, -3	Supported						
Kintex UltraScale +	-1LV	Not Supported		Supported				
	-1, -1L, -2LV	Not Supported	Supported					
	-2, -2L, -3	Supported						
Virtex UltraScale +	-1, -2LV	Not Supported	Supported					
	-2, -2L, -3	Supported						

Notes:

1. 5Kp100, 5Kp120 resolutions are supported only in 8ppc mode for -1LV device due to the s_axis_video_clock fmax restriction.
2. Maximum FRL rate supported by IP will be selected automatically based on device family and speed grade. You must provide the frl_clock frequency mentioned in *HDMI 2.1 Receiver Subsystem Product Guide* (PG351). However, you can lower the FRL rate by changing the MAX FRL rate in HF-VSDB block of EDID.

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).

Related Information

TX Initialization (TXI) Register (0x001C)
RX Initialization (RXI) Register (0x0024)

Interrupts

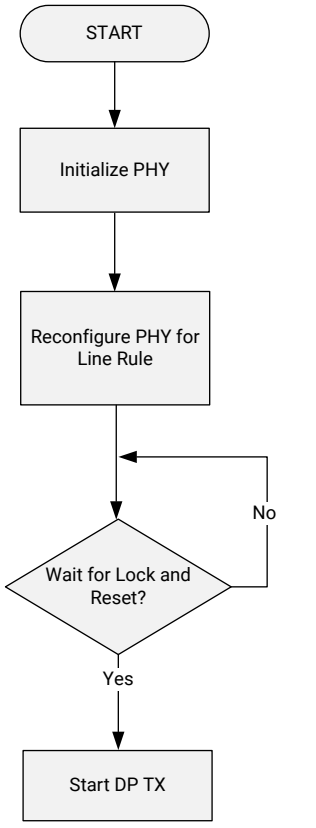
This core issues an active-High IRQ and must be used with an interrupt controller that is Level-High sensitive. Using sensitivity modes other than what was specified might result in incorrect software behavior.

Program and Interrupt Flow

Video PHY DisplayPort TX Flow

The following figure shows the Video PHY Controller core in the DisplayPort TX program flow.

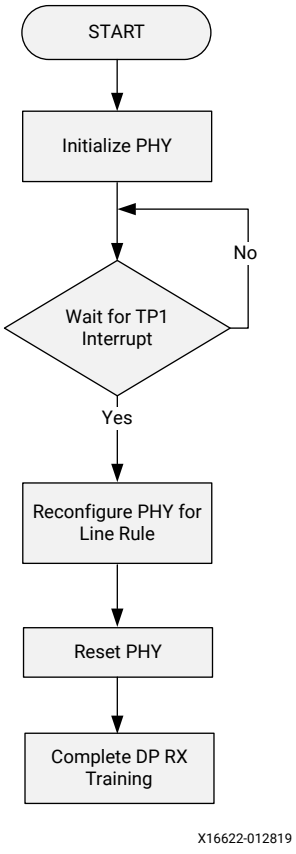
Figure 15: **Video PHY DisplayPort TX Program Flow**



Video PHY DisplayPort RX Flow

The following figure shows the Video PHY Controller core in the DisplayPort RX program flow.

Figure 16: **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 the following sections. At execution, the software application initializes the Video PHY Controller 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.

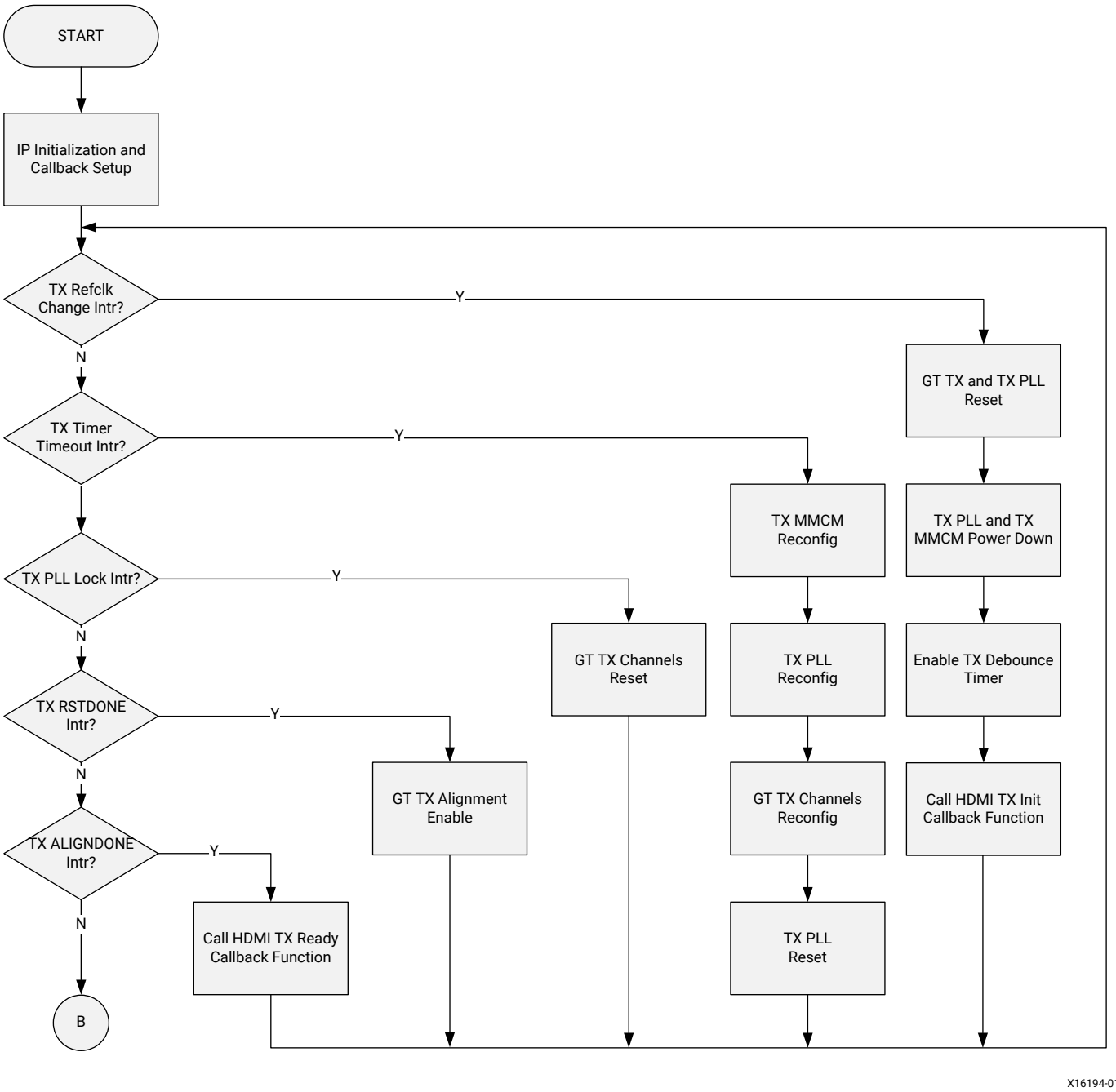
Note: The Video PHY Controller driver does not carry the video format, resolution, or color space information. Such information is handled by the HDMI TX and RX MAC. See the *HDMI 1.4/2.0 Transmitter Subsystem Product Guide* (PG235) and *HDMI 1.4/2.0 Receiver Subsystem Product Guide* (PG236) for more information.

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 or it can be forced by the setting the TX Frequency Reset bit (bit 3) of the Clock Detector Control register (0x200). Note that this bit is self clearing. See HDMI Reference Clock Requirements for details about `tx_refclk_rdy` port implementation.

There are several API callback hooks that the Video PHY Controller core executes 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 17: **HDMI TX Program Flow**



***Note*:** B in the previous figure continues on the next figure.

Related Information

HDMI Reference Clock Requirements

Video PHY Controller Core Driver TX Callbacks

The Video PHY Controller core driver TX callbacks are:

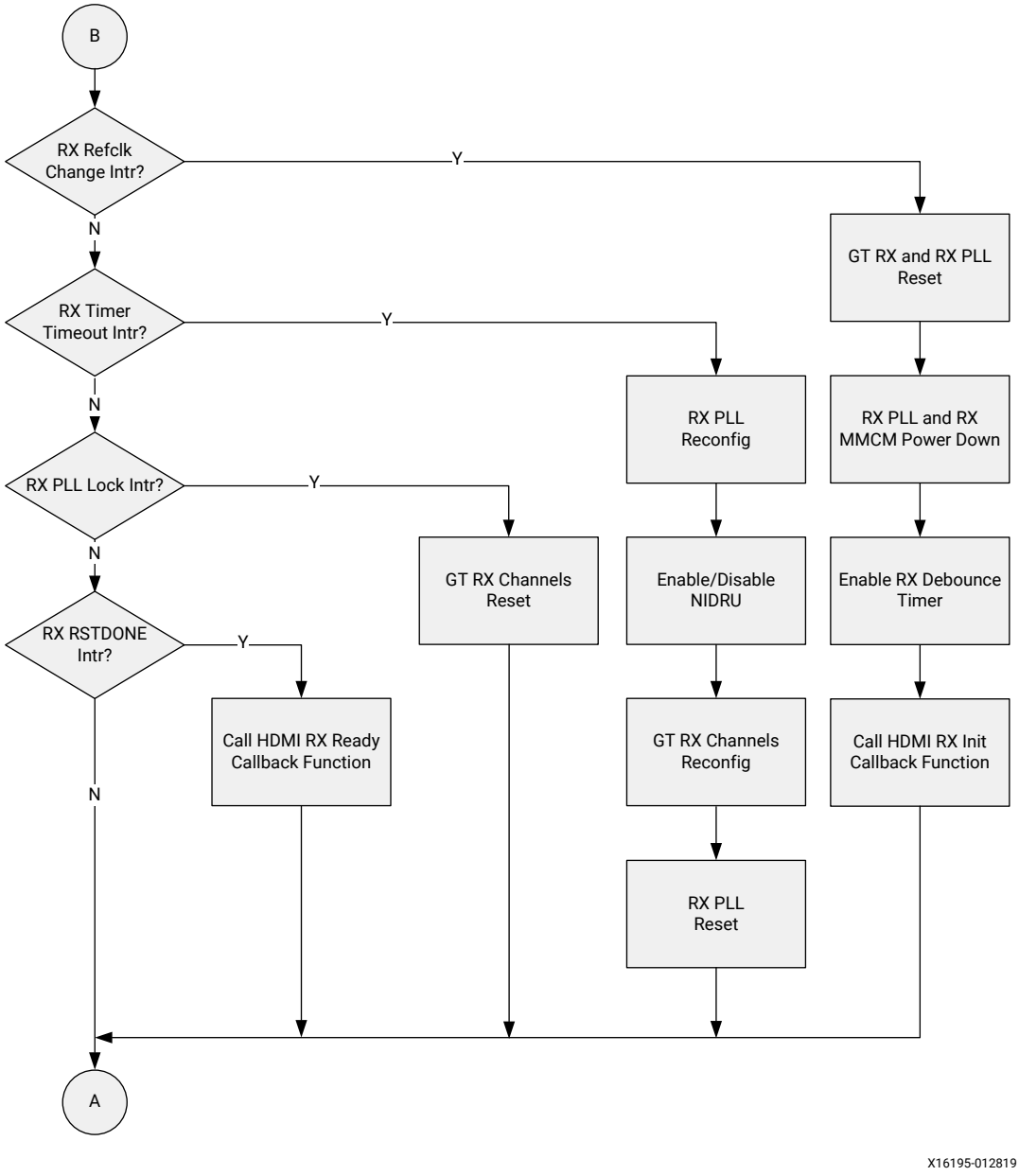
- **Video PHY Controller HDMI TX Init Callback:** This callback is named XVPHY_HDMI_HANDLER_TXINIT in the Video PHY Controller core driver. It is executed or called every time a change in the HDMI TX Transition Minimized Differential Signaling (TMDS) clock frequency occurs. This normally triggers a reset 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.

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 18: **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.

UltraScale GTHE3 and UltraScale+ GTHE4 and GTYE4 HDMI Implementation

The GTHE3 transceiver in the UltraScale™ FPGAs and the GTHE4 and GTYE4 transceivers in the UltraScale+™ FPGAs have 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 the 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 on 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.

CPLL Use

Because the largest multiplier that can be applied by the CPLL is 40, the minimum TMDS clock frequency that can be supported by the CPLL is 50 MHz. Video formats that have a TMDS clock frequency of less than 50 MHz are not supported by the CPLL.

Note: The largest multiplier (40) is achieved by CPLL CPLL_FBDIV_45 (N1)=5 and CPLL_FBDIV (N2)=8. Take note that CPLL_FBDIV=8 is only approved by Xilinx to be used in the Video PHY Controller IP for HDMI and DisplayPort protocols.

When the GT driver detects that the TMDS clock frequency is less than 50 MHz, it enables the NI-DRU to receive these low bit rates that are less than 500 Mb/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 50 MHz, a multiplier of 10X, 20X, or 40X is applied to keep the VCO frequency in the proper range as shown in the following table.

Table 77: **UltraScale GTH CPLL Use**

TMDS Clock Frequency (MHz)	CPLL Refclk Divider	CPLL Multiplier	VCO Frequency	Notes
<50	TX: Line Rate Dependent RX: 2	TX: Line Rate Dependent RX: 16	TX: Line Rate Dependent RX: 2.5 GHz	TX: Oversampling RX: NI-DRU is used
50 to 156.25	1	40	2.0 to 6.25 GHz	CDR 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.*

The following table is for illustration only. For unlisted color formats, support might be possible if they are not restricted by the VCO frequency range.

Table 78: **CPLL Support of RGB and YCbCr 4:4:4 Video Formats**

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

Table 78: **CPLL Support of RGB and YCbCr 4:4:4 Video Formats** *(cont'd)*

Resolution (Hz)	Bits Per Pixel			
	24	30	36	48
WXGA + 60	√	√	√	√
UXGA 60	√	√	√	√
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.
3. 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 with QPLL0 having the priority if the TMDS clock frequency falls within the valid range of QPLL0 and QPLL1. For the transmitter, the Video PHY Controller core driver uses oversampling and the dynamic QPLL switching to work around the QPLL limitations.

QPLL0 and QPLL1 Use

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 with QPLL0 having the priority if the TMDS clock frequency falls within the valid range of QPLL0 and QPLL1. For the transmitter, the Video PHY Controller core driver uses oversampling and the dynamic QPLL switching to work around the QPLL limitations.

When the GT driver detects that the TMDS clock frequency is less than 61.25 MHz, it enables the NI-DRU to receive these lower bit rates that are less than 0.6125 Gb/s. The NI-DRU runs at 2.5 Gb/s, which enables it to recover line rates that cannot be supported by the QPLL0. The Video PHY Controller does not use QPLL1 in NI-DRU mode.

The VCO of the QPLL0 must run in the frequency range of 9.8 GHz to 16.375 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.*

The following table shows how the TMDS clock frequency interacts with the QPLL0 and the frequency ranges that can be supported.

Table 79: **UltraScale GTH QPLL0 Use**

TMDS Clock Frequency (MHz)	QPLL0 Multiplier	Notes
<61.25	TX: Line Rate Dependent RX: 64	TX: Oversampling RX: NI-DRU is used
61.25 to 102.34375	160	Supported
102.34375 to 122.5	–	TMDS clock range cannot be supported
122.5 to 204.6875	80	Supported
204.6875 to 245	–	TMDS clock range cannot be supported
245 to 409.375	40	Supported
409.375 to 490	–	TMDS clock range cannot be supported

If the QPLL0 is used as the clock source, video formats with a TMDS clock of 102.34375 MHz to 122.5 MHz, 204.6875 MHz to 245 MHz, and frequencies higher than 409.375 MHz cannot be supported because 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 HDMI.*

The following table 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 80: **UltraScale GTH QPLL1 Use**

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

The following table is for illustration only. For unlisted color formats, support might be possible if they are not restricted by the VCO frequency range.

Table 81: **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	√	√	√	√
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	√	√	√	√
WUXGA 60	√	√	√	√
WSXGA 60	√	√	√	√

Notes:

1. This format is not supported in TMDS mode 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 HDMI Implementation

The GTX transceiver in 7 series FPGAs has two types of PLLs, the Quad PLL (QPLL) and the CPLL. The QPLL is shared by all four transceivers in the Quad. Each transceiver has its own CPLL. The Video PHY Controller 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 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. Take note that using bonded mode can only be done in the software by setting both TXSYSCLKSEL (and TXPLLCLKSEL) and RXSYSCLKSEL (and RXPLLCLKSEL) bits of RCS register to CPLL or QPLL. Setting the same PLL selection for TX and RX in the Video PHY Controller Vivado IDE causes a parameter validation error. While on bonded mode, note that the resolutions that require DRU cannot be transmitted by the Video PHY Controller core because the TX is being clocked by the DRU REFCLK (125 MHz). The TX REFCLK must run at its own clock which is TX TMDS clock * Oversampling Factor to properly transmit lower resolutions.

The QPLL and the CPLL both have certain limitations. The QPLL has "holes" which are 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 2.5 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 the following table.

Table 82: **7 Series GTX CPLL Usage**

TMDS Clock Frequency (MHz)	CPLL Refclk Divider	CPLL Multiplier	VCO Frequency	Notes
<80	TX: Line Rate Dependent RX: 2	TX: Line Rate Dependent RX: 20	TX: Line Rate Dependent RX: 2.5 GHz	TX: Oversampling RX: NI-DRU is used

Table 82: **7 Series GTX CPLL Usage** *(cont'd)*

TMDS Clock Frequency (MHz)	CPLL Refclk Divider	CPLL Multiplier	VCO Frequency	Notes
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. The following table shows the standard formats that are supported by the CPLL, indicating which require the DRU.

The following table is for illustration only. For unlisted color formats, support might be possible if they are not restricted by the VCO frequency range.

Table 83: **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	√	√	√
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	√	√	√

Table 83: **CPLL Support of RGB and YCbCr 4:4:4 Video Formats** *(cont'd)*

Resolution (Hz)	Bits Per Pixel			
wxga+p60	√	√	√	√
uxgap60	√	√	√	(2)
wuxgap60	√	√	√	√
wsxga+p60	√	√	√	√

Notes:

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

When the GT driver detects that the TMDS clock frequency is less than 74.125 MHz, it enables the NI-DRU to receive these lower bit rates that are less than 0.74125 Gb/s. The NI-DRU runs at 2.0 Gb/s, which enables it to recover line rates that cannot be supported by the QPLL.

The following table shows how the TMDS clock frequency interacts with the QPLL and the frequency ranges that can be supported.

Table 84: **7 Series GTX QPLL Usage**

TMDS Clock Frequency (MHz)	QPLL Multiplier	Notes
This format is not supported because it exceeds the<74.125	TX: Line Rate Dependent RX: 40	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
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 bands. For HDMI, the QPLL is always used in lower band because the upper band is not available in all speed grades.

The following table 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.

The following table is for illustration only. For unlisted color formats, support might be possible if they are not restricted by the VCO frequency range.

Table 85: **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	DRU
svgap60	DRU	DRU	DRU	DRU
xgap60	DRU	√	√	–
sxgap60	–	–	√	–
wxgap60	DRU	√	–	–
wxga+p60	√	–	–	√
uxgap60	√	–	–	(2)
wuxgap60	√	–	–	√
wsxga+p60	–	√	–	–

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 GTPE2 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 2.5 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 the following table.

Table 86: **Xilinx 7 Series GTP PLL0/1 Usage**

TMDS Clock Frequency (MHz)	PLL0/1 Refclk Divider	PLL0/1 Multiplier	VCO Frequency	Notes
<80	TX: Line Rate Dependent RX: 2	TX: Line Rate Dependent RX: 25	TX: Line Rate Dependent RX: 2.5 GHz	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 Gb/s.*

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. The following table shows the standard formats that are supported by the PLL0/1, indicating which require the DRU.

The following table is for illustration only. For unlisted color formats, support might be possible if they are not restricted by the VCO frequency range.

Table 87: **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	√	√	√	√
2160p60	-	(1)	(1)	(1)
vgap60	DRU	DRU	DRU[3]	DRU[3]
svgap60	DRU	DRU	DRU	√
xgap60	DRU	√	√	√
sxgap60	√	√	√	√
wxgap60	DRU	√	√	√
wxga+p60	√	√	√	√
uxgap60	√	√	√	(2)
wuxgap60	√	√	√	√
wsxga+p60	√	√	√	√

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.

Design Flow Steps

This section 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)
- *Vivado Design Suite User Guide: Designing with IP* (UG896)
- *Vivado Design Suite User Guide: Getting Started* (UG910)
- *Vivado Design Suite User Guide: Logic Simulation* (UG900)

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) 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:

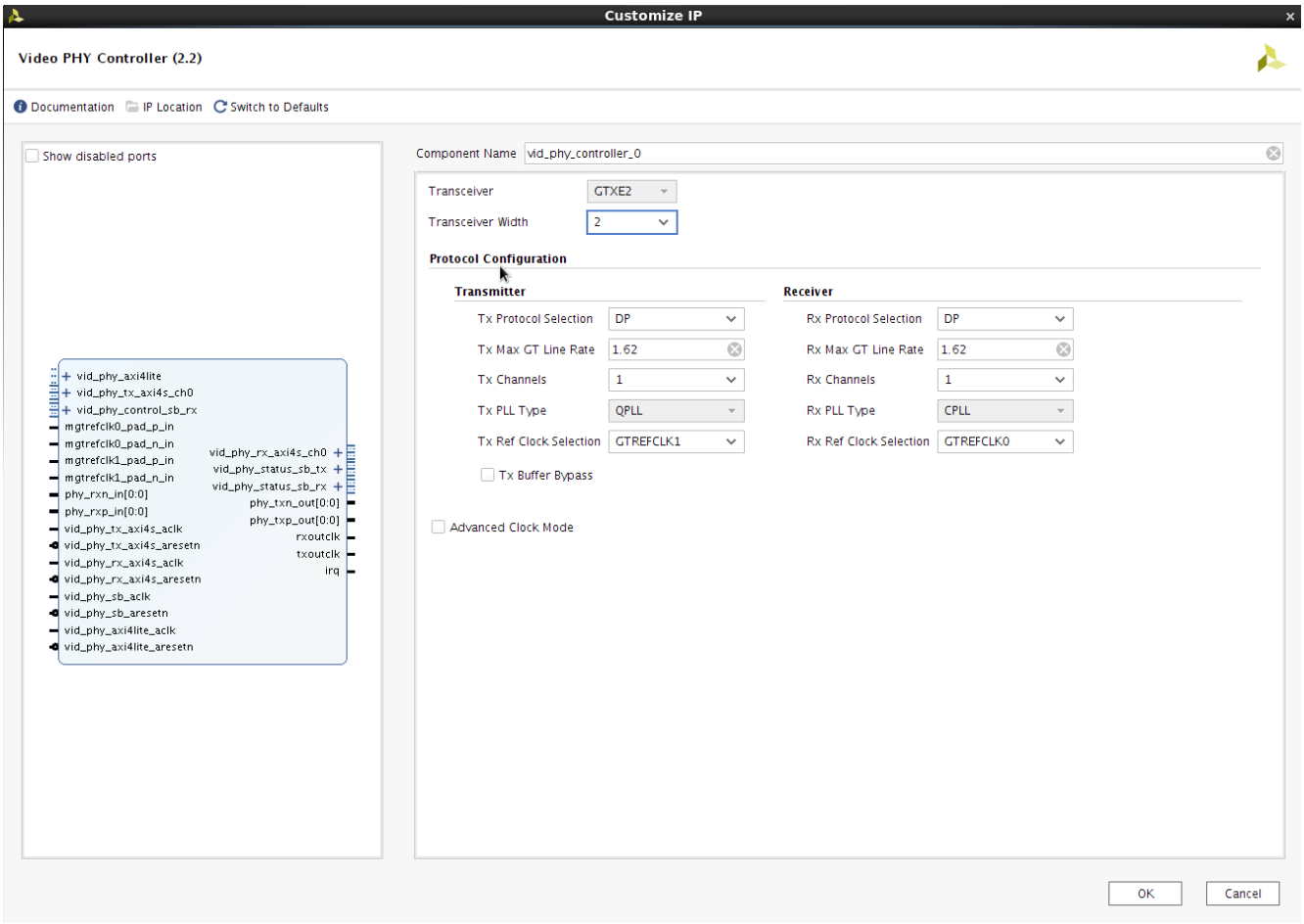
1. Select the IP from the IP catalog.
2. Double-click the selected IP or select the Customize IP command from the toolbar or right-click menu.

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

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

DisplayPort Protocol with GTXE2 Transceivers

Figure 19: **Vivado IDE – DisplayPort Protocol with GTXE2 Transceivers**

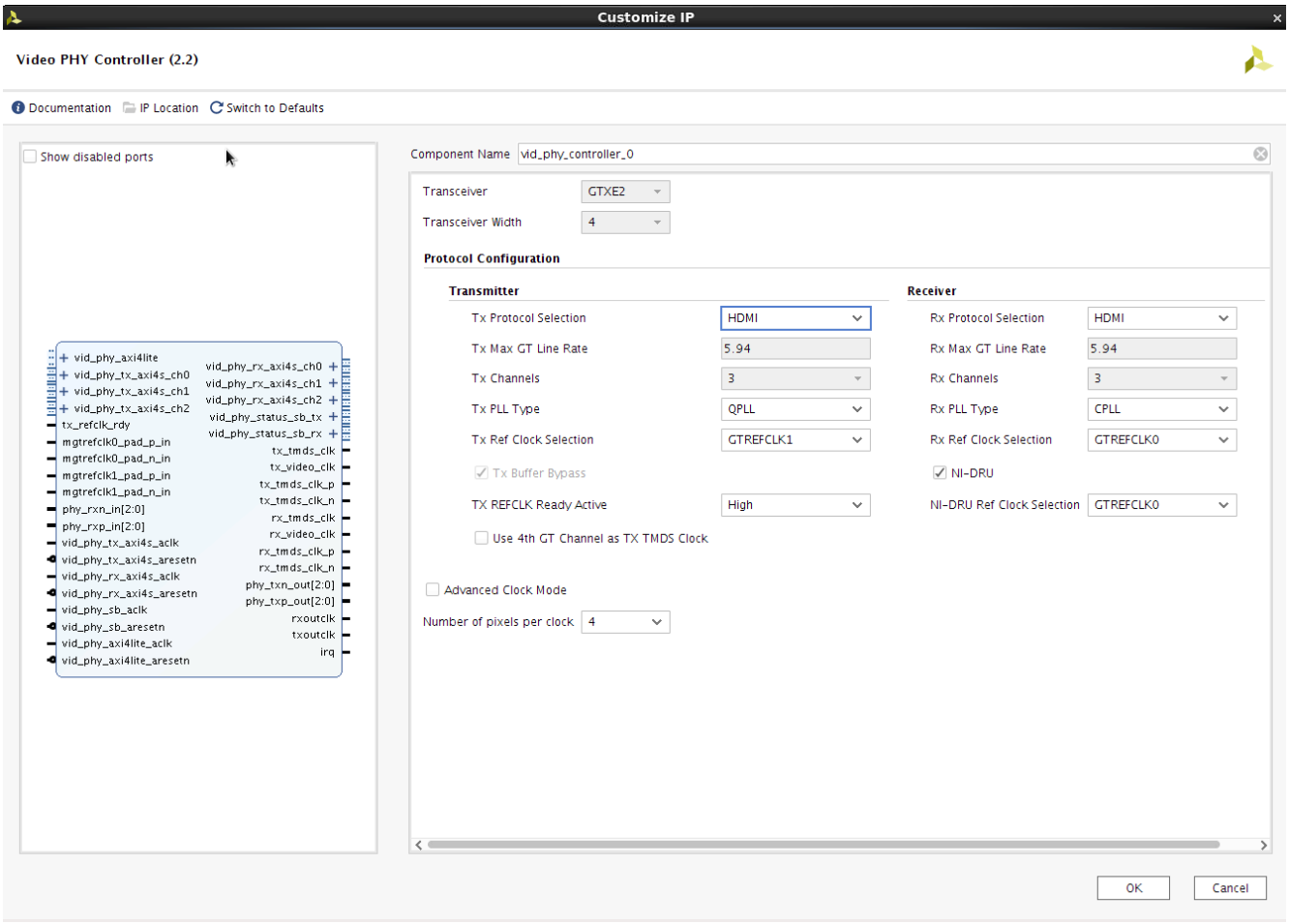


Selecting Ref Clock for Transmit or Receive Path indicates the default setup that is passed to GT Wizard reset state machines. You can also select different reference clock in application by programming Video PHY configuration registers and apply reset in appropriate paths.

***Note*:** Setting QPLL as the TX PLL type and CPLL as the RX PLL type is the valid configuration of the Video PHY Controller driver, based on the DisplayPort configuration and it is not recommended that you change the TX/RX PLL types. Hence, it is grayed out in the Vivado® IDE customization screen.

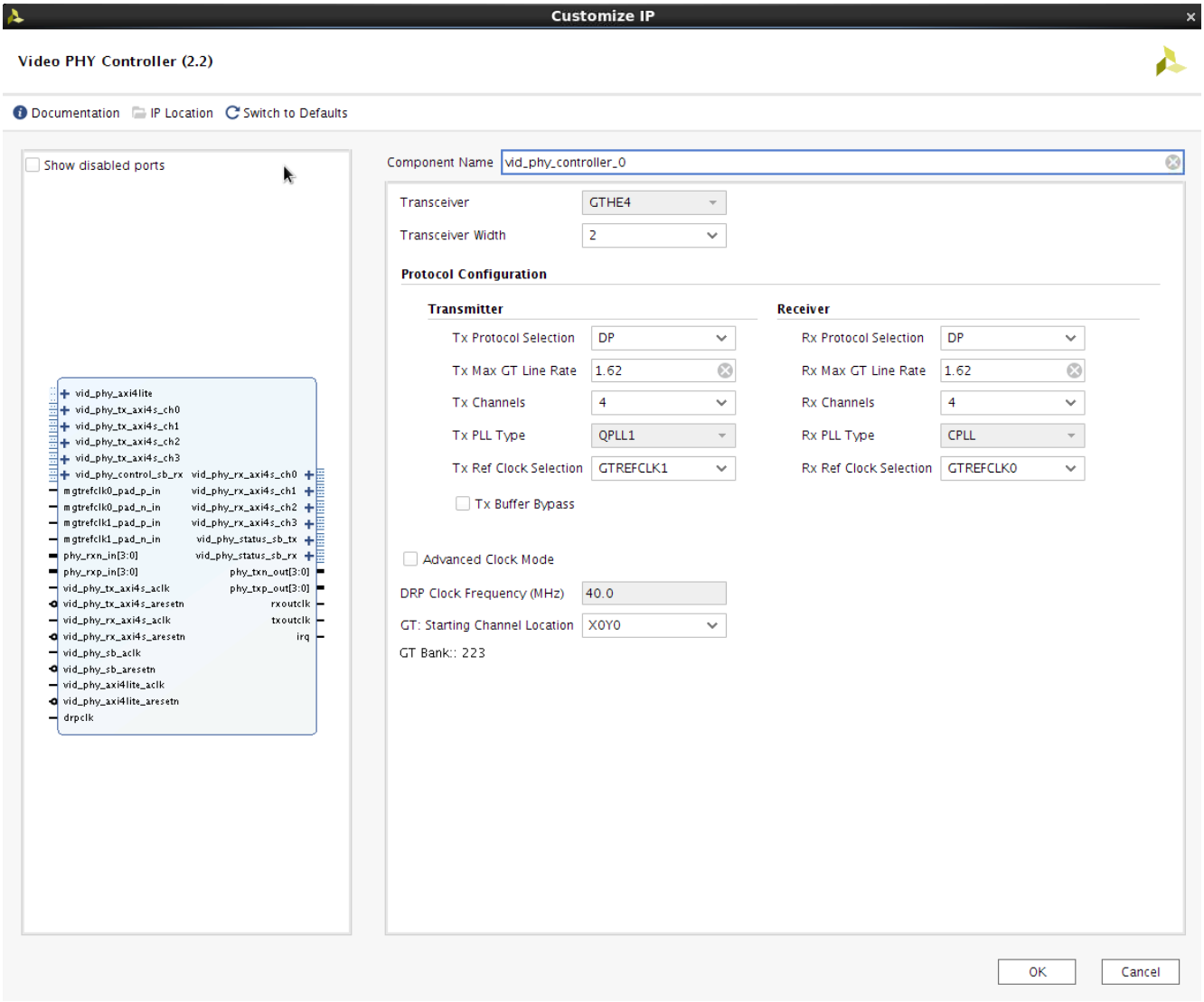
HDMI Protocol with GTPE2 and GTXE2 Transceivers

Figure 20: **Vivado IDE – HDMI Protocol with GTPE2 and GTXE2 Transceivers**



DisplayPort Protocol with GTHE3 and GTHE4 Transceiver

Figure 21: **Vivado IDE – DisplayPort Protocol with GTHE3 and GTHE4 Transceiver**

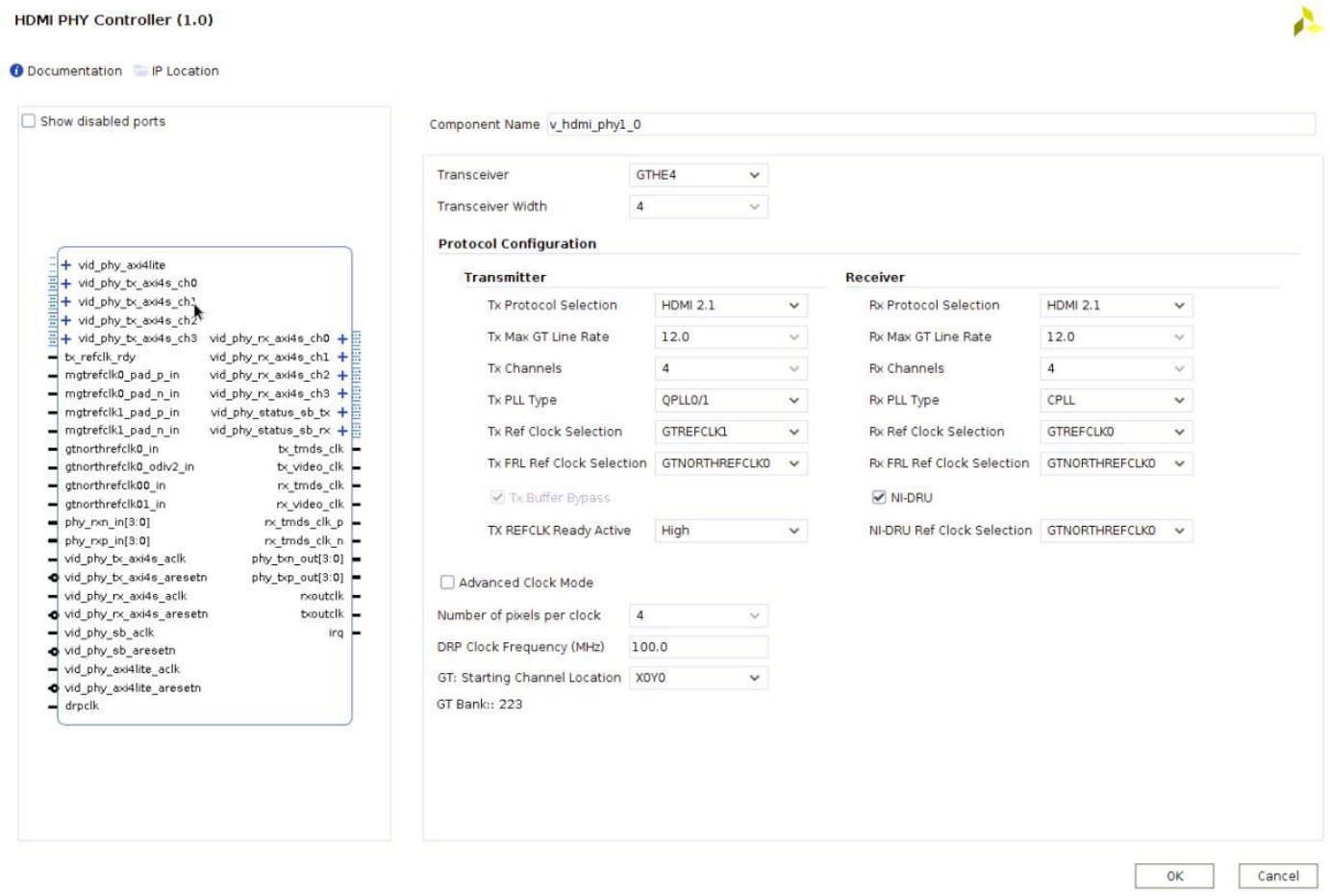


Selecting Ref Clock for Transmit or Receive Path indicates the default setup that is passed to GT Wizard reset state machines. You can also select different reference clock in application by programming Video PHY configuration registers and apply reset in appropriate paths.

***Note*:** Setting QPLL as the TX PLL type and CPLL as the RX PLL type is the valid configuration of the Video PHY Controller driver, and it is not recommended that you change the TX/RX PLL types. Hence, it is grayed out in the Vivado® IDE customization screen.

HDMI Protocol with GTHE3, GTHE4, and GTYE4 Transceivers

Figure 22: **Vivado IDE – HDMI 2.1 Protocol with GTHE3, GTHE4, and GTYE4 Transceivers**



Parameter Description

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. This option is not editable and depends on the FPGA family. The possible types are GTPE2, GTXE2, GTHE3, GTHE4, and GTYE4.

- **Transceiver Width:** Specifies the width of the transceiver that is used in this core.
- **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. For HDMI, it is important to note that the GT COMMON is optimized out of the Video PHY Controller when QPLL or QPLL0/1 is not associated with either the TX or the RX. This means that for GTXE2, dynamic switching from CPLL to QPLL is not possible when QPLL is optimized out from the design. Consider the following scenarios as examples:

  - GTHE3 with TX is DisplayPort and RX is None.
    - The only configurable option on 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 on RX 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 Gb/s.
- **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.
- **TX REFCLK Ready Active:** Specifies active-Low/High for TX RefClk Ready. This option is displayed when HDMI protocol is selected.
- **Use 4th GT Channel as TX TMDS Clock:** To enable/disable the function specifies as per the display option. This option is displayed when HDMI protocol is selected.
- **Ni-DRU:** When checked, the NI-DRU is included in the core. This option is displayed when HDMI protocol is selected for receiver.
- **Ni-DRU Ref Clock Selection:** Specifies the reference clock that corresponds to the NI-DRU. This option is displayed when HDMI protocol is selected for receiver.

IMPORTANT! *There is no automatic check between the DRU Ref Clock and the RX/TX Ref Clock Selection so you should avoid using the same clock for the DRU Ref Clock as either the TX or RX PLL Ref Clock.*

- **Number of pixels per clock:** Specifies the number of pixels for video clock generation. (Option is displayed when HDMI protocol is selected.
- **Advanced Clock Mode:** When checked under DisplayPort protocol, the core exposes all the available single-ended clock ports (for example, gtnorth/southrefclk_0/1 ports are revealed for GTXE2, GTHE3, GTHE4). When checked under HDMI protocol, the core exposes its active/selected single-ended clock ports and additional odiv_2 (only applicable for GTXE3 and GTHE4) clock ports. If unchecked, the core only exposes its active/selected differential clock ports.
- **DRP Clock Frequency (MHz):** Specifies the frequency that needs to be driven at the DRP clock. This option is displayed with a fixed value when the 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.

Related Information

DisplayPort Protocol with GTXE2 Transceivers
HDMI Protocol with GTPE2 and GTXE2 Transceivers
DisplayPort Protocol with GTHE3 and GTHE4 Transceiver
HDMI Protocol with GTHE3, GTHE4, and GTYE4 Transceivers

User Parameters

The following table shows the relationship between the fields in the Vivado® IDE and the user parameters (which can be viewed in the Tcl Console).

Table 88: **User Parameters**

Vivado IDE Parameter/ Value	User Parameter/Value	Default Value	Register Encoding
TX/RX Protocol Selection	C_Tx/Rx_Protocol	DP	
DP			
HDMI			
None			
TX/RX Max GT Line Rate	Rx_Max_GT_Line_Rate	8.1 for DP 1.4 for UltraScale and UltraScale+ devices 5.4 Gb/s for DP 1.2 5.94 for HDMI	

Table 88: **User Parameters** *(cont'd)*

Vivado IDE Parameter/ Value	User Parameter/Value	Default Value	Register Encoding
TX/RX Channels	C_Tx/Rx_No_Of_Channels	4 for DP 3 for HDMI	
Tx PLL Type[1]	C_TX_PLL_SELECTION	3 for transceiver GTPE2 and GTXE2 2 for DP under transceiver GTHE3, GTHE4, and GTYE4 6 for HDMI under transceiver GTHE3, GTHE4, and GTYE4	TX/RXSYSCLKSEL TX/RXPLLCLKSEL
CPLL: 0			00
QPLL0: 1			TX/RXSYSCLKSEL: 10 TX/RXPLLCLKSEL: 11
QPLL1: 2			TX/RXSYSCLKSEL: 11 TX/RXPLLCLKSEL: 10
QPLL: 3			11
PLL0: 4			00
PLL1: 5			11
QPLL0/1: 6			Follow QPLL0 or QPLL1 encoding
Rx PLL Type[1]	C_RX_PLL_SELECTION	0	Similar to TX PLL Type
Tx Ref Clock Selection[1]	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			001 010 011 100 101 110 011 100 101 110
Rx Ref Clock Selection[1]	C_RX_REFCLK_SEL	0	Similar to Tx Ref Clock Selection
Tx Buffer Bypass	Tx_Buffer_Bypass	TRUE for HDMI false for DisplayPort	
TX REFCLK Ready Active	C_Txrefclk_Rdy_Invert	false	
Use 4th GT Channel as TX TMDS Clock	C_USE_GT_CH4_HDMI	High	
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	

Table 88: **User Parameters** *(cont'd)*

Vivado IDE Parameter/ Value	User Parameter/Value	Default Value	Register Encoding
Number of pixels per clock Value Selection 1 2 4	C_INPUT_PIXELS_PER_CLOCK	4	
DRP Clock Frequency (MHz)	DRPCLK_FREQ	40 for DP 100 for HDMI	
Transceiver Width Value Selection 2 4	Transceiver_Width	2	
GT: Starting channel Location	CHANNEL_SITE	the lowest number from available X<num>Y<num> in GT	
Use ODDR/ODDRE1 for TX and RX differential TMDS clock out	C_Use_Oddr_for_Tmds_Clkout[3]	TRUE	
TX TMDS Clock output buffer 7 series: none, bufg, bufh, bufmr, bufr UltraScale/UltraScale+: none, bufg[4]	C_Tx_Tmds_Clk_Buffer[3]	none – for 7 series bufg – for UltraScale/UltraScale+	
TX Video Clock output buffer 7 series: none, bufg, bufh, bufmr, bufr UltraScale/UltraScale+: none, bufg[4]	C_Tx_Video_Clk_Buffer[3]	bufg	
RX TMDS Clock output buffer 7 series: none, bufg, bufh, bufmr, bufr UltraScale/UltraScale+: none, bufg[4]	C_Rx_Tmds_Clk_Buffer[3]	bufg	
RX Video Clock output buffer 7 series: none, bufg, bufh, bufmr, bufr UltraScale/UltraScale+: none, bufg[4]	C_Rx_Video_Clk_Buffer[3]	bufg	
TX Link Clock output buffer (7 series only) none, bufg, bufh, bufmr, bufr[4]	C_Tx_Outclk_Buffer[3]	bufg	
TX REFCLK input buffer to fabric (7 series only) none, bufg, bufh, bufmr, bufr[4]	C_Tx_Refclk_Fabric_Buffer[3]	bufg	
RX Link Clock output buffer (7 series only) none, bufg, bufh, bufmr, bufr[4]	C_Rx_Outclk_Buffer[3]	bufg	
DRU REFCLK input buffer to fabric (7-series only) none, bufg, bufh, bufmr, bufr[4]	C_Dru_Refclk_Fabric_Buffer[3]	none	

Table 88: **User Parameters** *(cont'd)*

Vivado IDE Parameter/ Value	User Parameter/Value	Default Value	Register Encoding
TX Phase Interpolator port enable (UltraScale and UltraScale+ only)	C_TXPI_Port_EN[1]	none	

Notes:

1. The Vivado IDE Parameter/Values are only used for the Video PHY Controller IP configuration in the Vivado IDE and are not the actual register encoding used to configure the Reference Clock Selection Register at offset 0x0010. The Video PHY Controller driver converts these parameters to the corresponding register encoding in the Register Encoding column.
2. The Vivado IDE Parameter/Values are only used for the Video PHY Controller IP configuration in Vivado environment and are not the actual register encoding used to configure the PLL REFCLKSEL bits (11:0) of the Reference Clock Selection Register at offset 0x0010. The Video PHY Controller driver converts these parameters to the corresponding register encoding in the Register Encoding column.
3. The user parameter applies to HDMI only and can be configured through Tcl command or through the Block Properties Window in IP integrator. Example:

```
set_property -dict [list CONFIG.C_Tx_Outclk_Buffer {none}] [get_ips <ip name>]
set_property -dict [list CONFIG.C_Rx_Video_Clk_Buffer {bufg}] [get_ips <ip name>]
set_property -dict [list CONFIG.C_Use_Oddr_for_Tmds_Clkout {false}] [get_ips <ip name>]
set_property -dict [list CONFIG.C_TXPI_Port_EN {true}] [get_ips <ip name>]
```

4. See Clocking.

Related Information

Clocking

Output Generation

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

Constraining the Core

Required Constraints

For GTHE3, GHE4, and GTYE4, the PHY generates an XDC file to place the GT channels and output data pins (optional 4th GT as TMDS). The PHY also constrains the clocks out of the GT. The required constrains include clock and location. The following clock ports are not constrained by the PHY, but are dependent on the application specific requirements:

- RX reference clock in
- TX reference clock in
- TX TMDS clock (when 4th GT is not used in TMDS)

- RX TMDS clock out

For more information, refer Chapter 6: Example Design.

For GTX2, the GT and output clock constraints are not created by the PHY and must be constrained by the system XDC file. See sample designs for examples of constraining the HDMI solution.

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

HDMI Reference Clock Requirements

The Video PHY Controller 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); the ports should be connected to a valid clock such as 50 MHz, 100 MHz, or 150 MHz clock. The system clock must be properly buffered (that is, BUFG) before it can be used and connected.

IMPORTANT! *Because the system clock is used by the Video PHY Controller as the reference clock for the frequency counter in its Clock Detector module, use an oscillator that has a jitter of less than ±40 PPM.*

The HDMI TX and RX reference clock (Transition Minimized Differential Signaling (TMDS) clocks) input frequency varies according to the input video and both are maximized at 297 MHz. Starting in Vivado 2018.3, the corresponding reference clock frequencies must be constrained at the Vivado Project top level XDC file at 297 MHz; that is, `create_clock -period 3.367 [get_ports <HDMI TX/RX REFCLK portname>]`. The NI-DRU reference clock frequency is fixed and is dependent on the transceiver type as follows:

- **GTXE2:** 125 MHz
- **GTHE3, GTHE4, and GTYE4:** 156.25 MHz

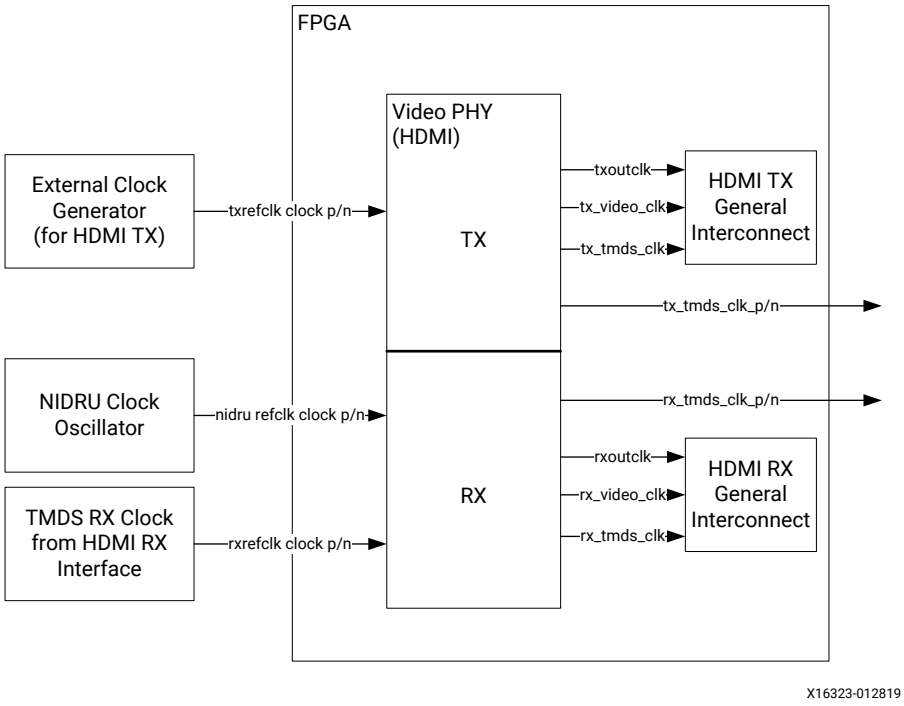
- **GTPE2:** 100 MHz

Starting in Vivado 2018.3, the NI-DRU reference clock frequency must be constrained at the Vivado Project top level XDC file at specified frequency; that is, for GTHE4: `create_clock -period 6.400 [get_ports <NI-DRU REFCCLK portname>]`

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.

The following figure illustrates the full reference clock requirement connections.

Figure 23: **Video PHY Controller HDMI Reference Clock Connections**



The HDMI TX reference clock comes from an external programmable clock generator capable of generating a range of frequencies from the minimum PLL reference clock (see HDMI TX Oversampled Reference Clock Requirements) to the maximum TMDS clock for supported video formats. For example, the Video PHY Controller TX uses the GTXE2 QPLL and supports 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, the Video PHY Controller uses the oversampling technique (see the following section for details).

The `txrefclk` port is accompanied by the `tx_refclk_rdy` port to indicate a lock condition. The `tx_refclk_rdy` port has three requirements:

- Connected to the external clock generator lock pin by default or can be toggled through GPIO. It must be toggled (deasserted then asserted) for every video format change. Alternatively, the TX Frequency Reset bit (bit 3) of the Clock Detector Control register (0x200) can be set if the `tx_refclk_rdy` port is active.
- It can *only* be asserted when the clock at `txrefclk_p/n` port is stable.
- At AXILITE CLK = 100 MHz, the `tx_refclk_rdy` minimum hold time are 5 us and 4 ms for fast switching and non-fast switching modes respectively.

IMPORTANT! *Failing to meet these requirements causes instability to the system.*

TX REFCLK frequency detection is sensitive only to the behavior of the `tx_refclk_rdy` port, a change on which triggers the clock detector to issue the TX frequency change event. The external clock generator is set up for the desired TX clock frequency, which means that the Video PHY Controller TX should get the requested frequency from the clock generator. Because the assumption is that Video PHY Controller 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 the TX. The RX is sensitive to the frequency change because there is no user control on the incoming RX TMDS clock.

Related Information

HDMI TX Oversampled Reference Clock Requirements

HDMI TX Oversampled Reference Clock Requirements

In normal cases, the GT reference clock requirement is equal to the TMDS clock requirement of a given HDMI resolution.

***Note*:** The GT reference clock requirement value can be accessed through the `HdmiTxRefClkHz` variable in the Video PHY Controller 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 Controller HDMI TX Flow). This value is ideally used in programming the external clock generator frequencies for GT TX operation.

The Video PHY Controller 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 Controller driver increases the reference clock by a factor of x3 or x5 until the minimum frequency for PLL is met. The following table shows the minimum reference clock per transceiver and PLL types.

For example, the GTHE3 CPLL needs to transmit 480p 60 Hz at eight bits per component. This video format requires a TMDS clock of 27 MHz which is below the GTHE3 CPLL 50 MHz minimum clock. The Video PHY Controller driver searches for the oversampling factor that satisfies this condition, which is x3. The new GT reference clock requirement is 81 MHz.

Table 89: **HDMI Transceiver to PLL Type Minimum Reference Clock**

Transceiver Type	PLL Type	HDMI Min Reference Clock (MHz)
GTHE3/GTHE4/GTYE4	QPLL0	61.25
	QPLL1	50
	CPLL	50
GTXE2	QPLL	74.125
	CPLL	80
GTPE2	PLL0/PLL1	80

Related Information

Video PHY DisplayPort RX Flow

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. The following table shows the frequency range needed from the clock generator if the Video PHY Controller is used to support all the video formats in the tables in the Video PHY Controller HDMI Implementation section.

Table 90: **External Clock Generator Typical Frequency Range**

Configuration	TX PLL	Reference Clock Range (MHz)
GTHE3/GTHE4/GTYE4	CPLL	50 to 297
	QPLL0	61.25 to 297
GTXE2	CPLL	80 to 297
	QPLL	74.125 to 297
GTPE2	PLL0/1	80 to 297

The following table shows the external clock generator frequency range if the Video PHY Controller 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 the GTHE3/GTHE4/GTYE4, GTXE2 CPLL, and GTPE2 PLL0/1 thus oversampling mode must be used to support it for the corresponding GT and PLL types.

Table 91: **External Clock Generator Frequency Range for SMPTE-SDI**

Transceiver Type	TX PLL	Reference Clock Range (MHz)	Remarks
GTHE3/GTHE4/GTYE4	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

Related Information

UltraScale GTHE3 and UltraScale+ GTHE4 and GTYE4 HDMI Implementation

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 the *HDMI 1.4/2.0 Transmitter Subsystem Product Guide* (PG235) 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 the *HDMI 1.4/2.0 Receiver Subsystem Product Guide* (PG236) 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 used. The constraints reflect the transceiver primitive site locations.

Starting in Vivado 2018.3, the MGT reference clock frequency must be constrained at the Vivado Project top level XDC file at specified frequency; that is, for GTHE4: `create_clock -period 6.400 [get_ports <NI-DRU REFCCLK portname>]`

Banking

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

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

Transceiver Placement

For 7 series devices, 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, the Vivado IDE provides customization for transceiver placement.

I/O Standard and Placement - DisplayPort

AUX Channel

The VESA DisplayPort Standard 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 devices supporting HR I/O 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 I/O 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 devices supporting HR I/O banks, use the following constraints:

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

For Virtex-7 devices supporting HP I/O 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.

I/O Standard and Placement - 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 I/O banks, use the following constraints:

I/O 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 I/O banks, use the following constraints:

I/O 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 I/O banks, use the following constraints:

I/O 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 I/O banks, use the following constraints:

I/O 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) I/O standard when the C_Use_GT_CH4_HDMI user parameters is set to false.

The RX TMDS and NI_DRU clock inputs are implemented as a GT reference clock input and therefore I/O 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:

I/O 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 the HDMI standard recommendations with proper level shifting or TMDS driver use.

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 I/O 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:

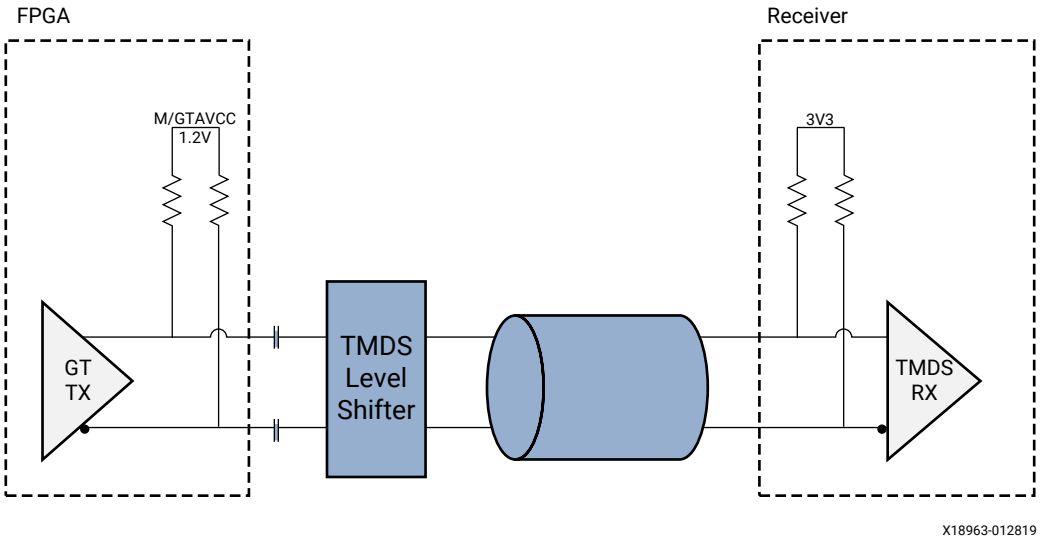
```
set_property PACKAGE_PIN E4 [get_ports {HDMI_RX_DAT_P_IN[0]}]
set_property PACKAGE_PIN D6 [get_ports {HDMI_RX_DAT_P_IN[1]}]
set_property PACKAGE_PIN B6 [get_ports {HDMI_RX_DAT_P_IN[2]}]
```

For UltraScale and UltraScale+ devices, actual pin assignments are absorbed by the GT Wizard instance in the Video PHY Controller thus pin assignment constraints are not required.

Board Design Guidelines - HDMI

Transmitter

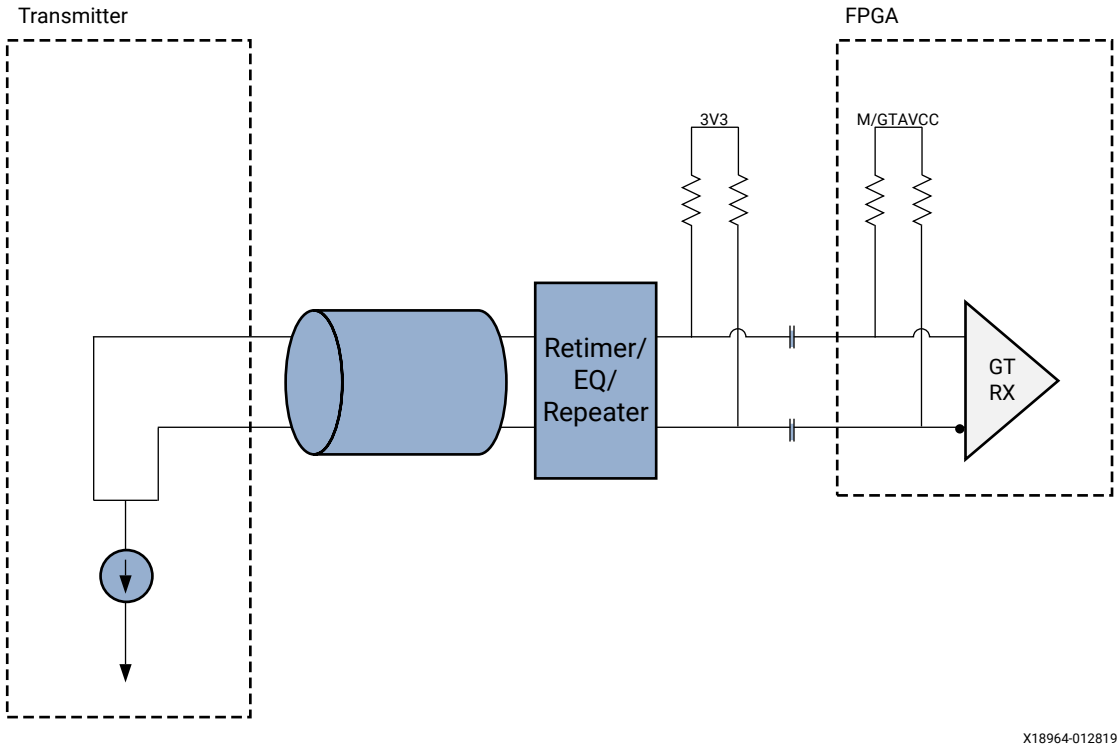
Figure 24: **Transmitter Board**



- The GT transmitter does 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
- Board design simulation must be done to ensure proper operation.
- The TI HDMI cable driver chip DP159 is used in all the example designs, including the evaluation board and the inrevium TB-FMCH-HDMI4K FMC daughter card. In the HDMI example design software, a sample DP159 driver is provided as reference. The settings have not been calibrated for various use cases. If you are using the DP159 in your product, you must adjust the settings based on the circuit design. You might need to perform some fine tuning depending on the compliance results.

Receiver

Figure 25: **Receiver Board**



- The GT Receiver does not support TMDS Level Signaling and must be used with a retimer or equalizer to be compliant with the TMDS specification. While the system can work without a TMDS retimer, Xilinx does not provide recommendations on this and recommend that all designs use a TMDS retimer.
- 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
- Board design simulation must be done to ensure proper operation.

Board Design Guidelines - DisplayPort

The DisplayPort receiver solution uses the DP159 or MCDP6000 retimer chips. Refer to the respective retimer datasheet and the DisplayPort Product guide for guidelines.

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).

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).

Example Design

HDMI Video PHY Controller Example Design

See the Application Software Development section of *HDMI 1.4/2.0 Transmitter Subsystem Product Guide* (PG235) and *HDMI 1.4/2.0 Receiver Subsystem Product Guide* (PG236) for details on running the HDMI example design flow.

***Note*:** The HDMI example design is only available for KC705, KCU105, ZC706, ZCU102, ZCU104, ZCU106, and VCU118 development boards.

***Note*:** Upgrading an example design created in Vivado 2017.4 or earlier to 2020.1 requires a software application update. This is especially the case for MMCM configurations using TX Buffer Bypass-enabled designs because of the migration from hardware to software.

DisplayPort Video PHY Controller Example Design

See the Example Design sections of *DisplayPort TX Subsystem Product Guide* (PG199), *DisplayPort RX Subsystem Product Guide* (PG233), *DisplayPort 1.4 TX Subsystem Product Guide* (PG299), and *DisplayPort 1.4 RX Subsystem Product Guide* (PG300) for details on running the DisplayPort Example Design flow.

***Note*:** The DisplayPort 1.2 example design is only available for KC705, KCU105, and ZCU102 development boards. The DisplayPort 1.4 example design is available for KCU105, ZCU102, and VCU118 development boards.

***Note*:** Upgrading the example design created in Vivado® Design Suite 2017.4 or earlier to 2018.1 requires a software application update. This is especially the case for MMCM configurations using TX Buffer Bypass-enabled designs because of the migration from hardware to software.

Chapter 7

Verification, Compliance, and Interoperability

Hardware Testing

For a list of tested boards, see the Example Design chapter.

Related Information

Example Design

XILINX

Appendix A

Upgrading

Parameter Changes

Parameter Changes

Table 92: **Video PHY Controller Changes**

V2.0	V2.1	V2.2	Note
Transceiver Width	No Change	Changed	In v2.2, this option is not configurable for HDMI.
Tx/Rx Max GT Line Rates	Changed	No Change	In v2.1, Display Port supports 8.1 Gb/s in GTHE3 and GTHE4.
Tx Channels	No Change	Changed	In v2.2, HDMI channel number used is 4 when the GT channel is enabled for Tx TMDS clock.
Not Applicable	TX REFCLK Ready Active	No Change	New option in v2.1.
Not Applicable	Not Applicable	Use 4th GT Channel as TX TMDS Clock	New option in v2.2.
Advanced Clock Mode	Changed	No Change	In v2.1, this option is available for HDMI.

Notes:

1. Not Applicable: GUI option is not available in this version.
2. Changed: GUI changes in this version (Refer to Note).
3. No Change: GUI option remain as in previous version.
4. GUI option names are in bold. See Customizing and Generating the Core for feature details.

Related Information

Customizing and Generating the Core

Port Changes

From v2.1 to v2.2, the following changes occurred:

- The HDMI TX and RX AXI4-Stream data width can be 20/40 bits depending on the transceiver width setting.

Register Changes

There is no register change from v2.0 to v2.1.

The following table shows the changes from v2.1 to v2.2.

Table 93: **Core Changes from v2.1 to v2.2**

Address	Register Name	Note
0x0070	TX Control (TXC)	Additional 1 bit for TXPRBSSEL per channel. (at bit - 6,14,22 and 30)
0x0100	RX Control (RXC)	Extended 1 bit extra for RXPRBSSEL register width
0x0110	Interrupt Enable Register (IER)	Added “TX MMCM Lock Change Event” and “RX MMCM Lock Change Event”
0x0114	Interrupt Disable Register (IDR)	Added “TX MMCM Lock Change Event“ and “RX MMCM Lock Change Event”
0x0118	Interrupt Mask Register (IMR)	Added “TX MMCM Lock Change Event” and “RX MMCM Lock Change Event”
0x011C	Interrupt Status Register (ISR)	Added “TX MMCM Lock Change Event” and “RX MMCM Lock Change Event”
0x0124	DRP Control MMCM TXUSRCLK	Newly introduced register for MMCM setting
0x0128	DRP CONTROL MMCM TXUSRCLK	Newly introduced register for MMCM setting
0x0144	DRP Control MMCM TRUSRCLK	Newly introduced register for MMCM setting
0x0148	DRP CONTROL MMCM RXUSRCLK	Newly introduced register for MMCM setting
0x0340	Control Register	Newly introduced register for TX TMDS Pattern Generator

Other Changes

From v2.0 to v2.1 of the core, the following changes occurred:

- Removed GTHE2 support for Display Port protocol.
- Added Tx Reference Clock Ready Invert support for HDMI protocol.
- Added 8.1 Gb/s maximum GT line rate support for Display Port protocol in GTHE3 and GTHE4.

From v2.1 to v2.2 of the core, the following changes occurred:

- Added GTYE4 support for HDMI and DisplayPort protocol.
- Removed GTPE2 support for Display Port protocol.
- Added 2 byte support for HDMI.
- Added GT channel support for TMDS clock in HDMI.

Software Driver - Deprecated APIs

- `XVphy_DrpRead` replaced by `XVphy_DrpRd`

  In Vivado 2017.3, the `XVphy_DrpRead` API was deprecated and replaced by `XVphy_DrpRd` to facilitate better status returns. In `XVphy_DrpRd`, the DRP readout return was separated from the XST_FAILURE/XST_SUCCESS return to easily identify if a certain DRP access failed. A new argument `*RetVal` was added to hold the DRP readout value. The following code example illustrates how to migrate from `XVphy_DrpRead` to `XVphy_DrpRd`.

  From `XVphy_DrpRead`:

  ```
  u16 DrpVal;
  DrpVal = XVphy_DrpRead(InstancePtr, QuadId, ChId, 0x63);
  ```

  To `XVphy_DrpRd`:

  ```
  u16 DrpVal;
   u32 Status = XST_SUCCESS;
   Status = XVphy_DrpRd(InstancePtr, QuadId, ChId, 0x63, &DrpVal);
  ```

  Note: Status holds XST_FAILURE/XST_SUCCESS.

  Note: `DrpVal` holds the DRP Readout value.

- `XVphy_DrpWrite` replaced by `XVphy_DrpWr`

  In Vivado 2017.3, the `XVphy_DrpWrite` API was deprecated and replaced by `XVphy_DrpWr` to align with the naming convention of `XVphy_DrpRd`. There is no functional difference between the two APIs, and you can replace each `XVphy_DrpWrite` function call with `XVphy_DrpWr`.

  From `XVphy_DrpWrite`:

  ```
  u16 DrpVal;
   u32 Status = XST_SUCCESS;
   Status = XVphy_DrpWrite(InstancePtr, QuadId, ChId, 0x63, DrpVal);
  ```

  To `XVphy_DrpRd`:

  ```
  u16 DrpVal;
   u32 Status = XST_SUCCESS;
   Status = XVphy_DrpWr(InstancePtr, QuadId, ChId, 0x63, DrpVal);
  ```

  Note: Status holds XST_FAILURE/XST_SUCCESS.

Note: `DrpVal` holds the value to write to the DRP address.

- `XVphy_HdmiInitialize` replaced by `XVphy_Hdmi_CfgInitialize`

  In Vivado 2017.3, the `XVphy_HdmiInitialize` API was deprecated and replaced by `XVphy_Hdmi_CfgInitialize` to provide ease-of-use in handling the VPHY system frequency clock.

  From `XVphy_HdmiInitialize`:

```
#define XPAR_CPU_CORE_CLOCK_FREQ_HZ 100000000
u32 Status;
Status = XVphy_HdmiInitialize(&Vphy, 0,
XVphyCfgPtr,XPAR_CPU_CORE_CLOCK_FREQ_HZ);
```

  To `XVphy_Hdmi_CfgInitialize`:

```
u32 Status;
Status = XVphy_Hdmi_CfgInitialize(&Vphy, 0, XVphyCfgPtr);
```

  Note: Status holds XST_FAILURE/XST_SUCCESS.

Appendix B

Debugging

This appendix includes details about resources available on the Xilinx® Support website and debugging tools.

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

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

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

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. The Xilinx Community Forums are also available where members can learn, participate, share, and ask questions about Xilinx solutions.

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.

Known Issues

Clock override critical warnings on `TXOUTCLK` or `RXOUTCLK` GT pins when compiling the DisplayPort and HDMI Video PHY Controller are known issues and can be ignored. This is necessary in constraining the GT pins per target maximum line rate. This is not normally a recommended flow but has been done for this specific case. The following is an example of the critical warning:

1.
```
[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]
```

2. Primary Clock declaration from the RX_CLKBUF_INST critical warnings on GTPE2 can be ignored. This is not normally a recommended flow but has been done for this specific case. Below is an example of the critical warning:

```
A primary clock my_ip_dru_clk_b0gt0 is created on an inappropriate pin
DUT/inst/RX_LCLK_BUF_INST/O. It is recommended to create a primary clock
only on a proper clock source (input port or primitive output pin with
no timing arc)

A primary clock my_ip_dru_clk_b0gt1 is created on an inappropriate pin
DUT/inst/RX_LCLK_BUF_INST/O. It is recommended to create a primary clock
only on a proper clock source (input port or primitive output pin with
no timing arc)

A primary clock my_ip_dru_clk_b0gt2 is created on an inappropriate pin
DUT/inst/RX_LCLK_BUF_INST/O. It is recommended to create a primary clock
only on a proper clock source (input port or primitive output pin with
no timing arc)

A primary clock my_ip_lclk_from_rxpll is created on an inappropriate pin
DUT/inst/RX_LCLK_BUF_INST/O. It is recommended to create a primary clock
only on a proper clock source (input port or primitive output pin with
no timing arc)
```

Note: TIMING-2 on GTP is by design. TIMING-17 is by design and can be ignored. The following is an example of the critical warning.

```
TIMING-17#1 Critical Warning
Non-clocked sequential cell
The clock pin DUT/inst/tx_tmdsclk_patgen_inst/txdata_counter_reg[0]/C is
not reached by a timing clock
```

New

Previous

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 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 Core

AR 57842.

Technical Support

Xilinx provides technical support on the Xilinx Community Forums 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 ask questions, navigate to the Xilinx Community Forums.

Debug Tools

There are many tools available to address Video PHY Controller design issues. It is important to know which tools are useful for debugging various situations.

Vivado Design Suite Debug Feature

The Vivado® Design Suite debug feature inserts logic analyzer and virtual I/O cores directly into your design. The debug feature also allows you to set trigger conditions to capture application and integrated block port signals in hardware. Captured signals can then be analyzed. This feature in the Vivado IDE is used for logic debugging and validation of a design running in Xilinx® devices.

The Vivado logic analyzer is used to interact with the logic debug LogiCORE IP cores, including:

- ILA 2.0 (and later versions)
- VIO 2.0 (and later versions)

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

Reference Boards

Various Xilinx® development boards support the Video PHY Controller 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
- Zynq®-7000 SoC ZC706
- Virtex® UltraScale+™ FPGA VCU118 Evaluation Kit
- Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit
- Zynq UltraScale+ MPSoC ZCU104 Evaluation Kit
- Zynq UltraScale+ MPSoC ZCU106 Evaluation Kit

Interface Debug

AXI4-Lite Interfaces

To verify that the interface is functional, read from a register that does not have all 0s as a default. 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 `vid_phy_axi4lite_aclk` input is connected and toggling.

- The interface is not being held in reset, and `vid_phy_axi4lite_aresetn` 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.

AXI4-Stream Interfaces

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

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

HDMI Debugging

- What to check if I do not see TX or RX Frequency Events?
  1. Make sure HDMI cable is properly inserted.
  2. Ensure that the correct GTREFCLK input pins are connected according to Customizing and Generating the Core.
  3. Ensure that cable detect and HPD pins are properly connected with correct active level setting in the HDMI TX or RX subsystem GUI.
  4. Try connecting with a different cable.
- Why is not there any video output when GTXE2 Video PHY Controller is on bonded mode and receiving with DRU?

  This is a known limitation of bonded mode.

  While on bonded mode, resolutions that require DRU cannot be transmitted by the Video PHY Controller because TX is being clocked by the DRU REFCLK (125 MHz). The TX REFCLK must run at TX TMDS clock * Oversampling Factor to properly transmitted low resolutions.
- How to debug if PLL gets stuck on reset?
  1. Check the RCS register and ensure the clock selections are done correctly.
  2. Check the GT initialization sequence after configuration.

For example, GTH GTTXRESET should be held HIGH until PLL_LOCK is asserted. Failing to do so might cause the PLL and GT to get stuck. See the *7 Series FPGAs GTX/GTH Transceivers User Guide* (UG476), *UltraScale Architecture GTH Transceivers User Guide* (UG576), and *7 Series FPGAs GTP Transceivers User Guide* (UG482) for more information.

3. Ensure the GTREFCLK is present and is driving the PLL.
4. Try toggling the PLL_GT_RESET bits of TXI or RXI registers.
5. Ensure that the Video PHY Controller driver and IP versions are from the same Vivado build.

- Why does the Video PHY Controller log shows error saying no DRU instance?

  This indicates that HDMI design received a video carrying a TMDS Clock that is below the PLL thresholds and there is no NI-DRU in the Video PHY Controller instance to receive it.

  The DRU must be enabled from the Video PHY Controller GUI to be able to receive resolutions below the PLL threshold. The DRU must be supplemented with a corresponding GT clock based on the requirements listed in Video PHY Controller HDMI Reference Clock Requirements (link below).

- Why do I see DRU reference clock frequency equal to 1 Hz?

  This happens when the clock detector module identifies a mismatch between the actual and required DRU REFCLK frequency. This can be due to two reasons:

  - The DRU clock frequency is outside ±10 kHz tolerance of the required frequency.
  - When the DRU Ref lock Selection is same as the RX or TX Ref Clock Selection, disabling the RX or TX Ref Clock Selection can also disable the DRU Ref Clock, which reports the DRU clock frequency as equal to 1 Hz.

- For GTXE2, when to switch in/out of bonded mode?

  Switch to CPLL bonded mode when transmitting or receiving videos that fall into QPLL holes.

  TIP: *The application reports any Video PHY Controller error: QPLL config not found*

  Switch out of bonded mode when operating in Passthrough mode and receiving DRU line rate.

  TIP: *The application reports Video PHY Controller Error: TX cannot be used on DRU & bonded mode*

  You can add functions inside the VphyProcessError API of `xhdmi_example.c` to automatically switch in and out of bonded mode depending on the error type.

- How to determine if a resolution falls into PLL hole?

  PLL holes are generally limitation of GTXE2 QPLL, when a video format's TMDS clock falls outside the supported range of the QPLL. Details of QPLL hole discussion can be found in the 7 series GTXE2 Video PHY Controller core HDMI Implementation section.

Related Information

Customizing and Generating the Core
7 Series GTXE2 HDMI Implementation
HDMI Reference Clock Requirements

TX Only Sample Video PHY Controller Log

The following log entries were taken from a GTXE2 Video PHY Controller using CPLL to clock the TX. The first log shows a typical flow for a video format change while the second one shows a log that attempted to retry the TX alignment.

TX Alignment retry is only applicable for GTXE2 devices and can occur more than once in a single format change. TX Alignment retry is attempted when the watchdog timer for TX alignment times out after the TX Reset Done state is attained.

```
VPHY log
------
TX frequency event
CPLL lost lock
TX frequency event
TX timer event
TX MMCM reconfig done
CPLL reconfig done
GT TX reconfig start
GT TX reconfig done
CPLL lock
TX MMCM lock
TX reset done
TX alignment done
```

```
VPHY log
------
TX frequency event
CPLL lost lock
TX frequency event
TX timer event
TX MMCM reconfig done
CPLL reconfig done
GT TX reconfig start
GT TX reconfig done
CPLL lock
TX MMCM lock
TX reset done
TX alignment watchdog timed out.
TX reset done
TX alignment done
```

***Note*:** TX MMCM lock is only for GTXE2.

RX Only Sample Video PHY Controller Log

The following log entry was taken from a GTXE2 Video PHY Controller using the QPLL to clock the RX. The log shows a typical flow for a video format change on the RX.

```
VPHY log
------
RX frequency event
RX timer event
RX DRU disable
QPLL reconfig done
GT RX reconfig start
GT RX reconfig done
QPLL lock
RX reset done
RX MMCM reconfig done
```

The following log entry shows the flow necessary to use the DRU clock and CPLL:

```
VPHY log
------
GT init start
GT init done
RX frequency event
RX timer event
DRU enable
CPLL reconfig done
GT RX reconfig start
GT RX reconfig done
CPLL lock
RX reset done
RX MMCM reconfig done
```

Passthrough Sample Video PHY Controller Log

The following log entry is from a GTXE2 Video PHY Controller using QPLL and CPLL to clock RX and TX respectively. The log shows a typical flow for a video format change.

```
VPHY log
------
RX frequency event
RX timer event
RX DRU disable
QPLL reconfig done
GT RX reconfig start
GT RX reconfig done
QPLL lock
RX reset done
RX MMCM reconfig done
TX frequency event
CPLL lost lock
TX frequency event
TX timer event
TX MMCM reconfig done
CPLL reconfig done
```

```
GT TX reconfig start
GT TX reconfig done
CPLL lock
TX MMCM lock
TX reset done
TX alignment done
```

Appendix C

Application Software Development

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

XILINX

Appendix D

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 (DocNav) provides access to Xilinx documents, videos, and support resources, which you can filter and search to find information. To open 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 DocNav, click the **Design Hubs View** tab.
- On the Xilinx website, see the Design Hubs page.

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

References

These documents provide supplemental material useful with this guide:

1. *VESA DisplayPort Standard* (VESA website) (1.2 and 1.4)

Vivado Documentation

1. *Vivado Design Suite User Guide: Designing IP Subsystems using IP Integrator* (UG994)
2. *Vivado Design Suite User Guide: Designing with IP* (UG896)
3. *Vivado Design Suite User Guide: Getting Started* (UG910)
4. *Vivado Design Suite User Guide: Logic Simulation* (UG900)
5. *Vivado Design Suite User Guide: Programming and Debugging* (UG908)
6. *Vivado Design Suite User Guide: Implementation* (UG904)

Xilinx Video IP Documentation

1. *DisplayPort TX Subsystem Product Guide* (PG199)
2. *DisplayPort RX Subsystem Product Guide* (PG233)
3. *HDMI 1.4/2.0 Transmitter Subsystem Product Guide* (PG235)
4. *HDMI 1.4/2.0 Receiver Subsystem Product Guide* (PG236)
5. *DisplayPort 1.4 TX Subsystem Product Guide* (PG299)
6. *DisplayPort 1.4 RX Subsystem Product Guide* (PG300)

Xilinx Device Documentation

1. *Virtex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS893)
2. *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS892)
3. *Kintex-7 FPGAs Data Sheet: DC and AC Switching Characteristics* (DS182)
4. *Virtex-7 FPGAs Data Sheet: DC and AC Switching Characteristics* (DS183)
5. *Zynq UltraScale+ MPSoC Data Sheet: DC and AC Switching Characteristics* (DS925)
6. *Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics* (DS181)
7. *Zynq-7000 SoC (Z-7030, Z-7035, Z-7045, and Z-7100) Data Sheet: DC and AC Switching Characteristics* (DS191)
8. *Kintex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics* (DS922)
9. *Virtex UltraScale+ FPGA Data Sheet: DC and AC Switching Characteristics* (DS923)

Xilinx Transceiver Documentation

1. *UltraScale Architecture GTY Transceivers User Guide* (UG578)
2. *UltraScale Architecture GTH Transceivers User Guide* (UG576)
3. *UltraScale Architecture Clocking Resources User Guide* (UG572)

4. *7 Series FPGAs GTX/GTH Transceivers User Guide* (UG476)
5. *7 Series FPGAs GTP Transceivers User Guide* (UG482)
6. *7 Series FPGAs Clocking Resources User Guide* (UG472)

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
06/30/2021 v2.2	
IP Facts Chapter 3: Product Specification	Updated notes. Updated the table RX Initialization (RXI) Register (0x0024) Updated the table Clock Detector (HDMI) Control Register (0x0200)
08/07/2020 v2.2	
GTXE2*/GTHE3 Clocking When Transmit Buffer Bypass is Disabled GTXE2*/GTHE3 Clocking When Transmit Buffer Bypass is Enabled	Clarifications added.
Using Fourth GT Channel as TX TMDS Clock Source	Recommendation note added.
Required Constraints	Updated constraints information for PHY.
12/06/2019 v2.2	
MMCM RXUSRCLK Control/Status (MMCM_RXUSRCLK_CTRL) Register (0x0140)	Bit 12 added
DisplayPort Clocking	Clocking tables updated Important note updated
GTXE2*/GTHE3 Clocking When Transmit Buffer Bypass is Enabled	BUFG_GT added to figure
DisplayPort Protocol with GTXE2 Transceivers DisplayPort Protocol with GTHE3 and GTHE4 Transceiver	Clarification note added
05/22/2019 v2.2	
TX TMDS Pattern Generator Control Register (0x0340)	Updated TX TMDS bits
UltraScale GTHE3 and UltraScale+ GTHE4 and GTYE4 HDMI Implementation	Update Bits Per Pixel=48 (UXGA 60) in Video Formats table
HDMI Reference Clock Requirements	Reference clock constraint updated
Clock Placement	Reference clock constraint updated
Board Design Guidelines - HDMI	GT receiver guidance updated
Known Issues	Primary clock critical warnings note added
12/05/2018 v2.2	
	• Added support for GTYE4 support in DisplayPort Clocking section • Added links to device datasheets • Added links to example designs

Section	Revision Summary
04/04/2018 v2.2	
	• Added UltraScale+™ GTYE4 support • Enabled 4KP24/25/30 10 BPC support in UltraScale+™/ UltraScale+™+ CPLL • Enabled usage of fourth GT channel as TX TMDS Clock source for HDMI • Converted MMCM control to DRP based accesses
11/29/2017 v2.0	
	• Updated to remove GTPE2 support for DisplayPort
10/04/2017 v2.1	
	• Updated to remove GTHE2 support
07/14/2017 v2.0	
	• Updated Artix®-7 support in IP Facts
04/05/2017 v2.0	
	• Added Clocking section • Added Video PHY HDMI Generated Clocks section • Updated Bank section • Updated I/O Standard and Placement section
06/08/2016 v2.0	
	• Updated for GTHE4 support
04/06/2016 v2.0	
	• 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 Version 2.0	
Initial release.	

Please Read: Important Legal Notices

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

AUTOMOTIVE APPLICATIONS DISCLAIMER

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

Copyright