XAPP1339 (v1.0) October 31, 2018

Implementing 2.5G MIPI D-PHY Controllers

Author: Jinhua Li, John Hu, and Tom Li

Summary

This application note provides an FPGA implementation of a high-speed mobile industry processor interface (MIPI) D-PHY solution for 2.5G. The solution uses GTH transceivers in the UltraScale™ or UltraScale+™ FPGA families and SelectIO™ technology, with simple external circuits to comply with the *MIPI Alliance specification for D-PHY*, version 2.1 [Ref 1]. The application note is based on the GTH transceiver in UltraScale FPGAs, and describes how to implement the solution in depth. This MIPI D-PHY solution is also applicable to UltraScale+ FPGAs. To implement this solution with other Xilinx transceiver technology, or to exceed the 2.5G design, contact the authors to reevaluate your scheme. This application note assumes that you are familiar with *UltraScale Architecture GTH Transceivers User Guide* (UG576) [Ref 2], *MIPI D-PHY v4.1 Product Guide* (PG202) [Ref 3], and *D-PHY Solutions* (XAPP894) [Ref 4].

IMPORTANT: *This application note is targeted at applications with line rates ranging from 100 Mb/s to 2.5 Gb/s. For lower speed applications, such as less than 1.5 Gb/s, refer to MIPI D-PHY v4.1 Product Guide (PG202) and D-PHY Solutions (XAPP894).*

Porting the Design from UltraScale to UltraScale+ FPGAs describes in detail the design method and the differences between the UltraScale and UltraScale+ FPGA GTH transceiver implementations.

Download the reference design files for this application note from the Xilinx website. For detailed information about the design files, see Reference Design.

Introduction

MIPI is a serial communication interface specification promoted by the MIPI alliance. An FPGA MIPI implementation provides a standard connection medium for cameras and displays referred to as a camera serial interface (CSI) or display serial interface (DSI). Both interface standards use the D-PHY specification, which provides a flexible, low-cost, high-speed serial interface solution.

Most Xilinx FPGAs do not yet have I/O that can natively support D-PHY, except in the UltraScale+ FPGAs. Connecting MIPI-equipped camera and display components requires implementing the D-PHY hardware specification with discrete components outside the FPGA lane side (see Figure 1). A design that functions as the lane control logic of the D-PHY as shown in Figure 1 can be implemented inside the FPGA.

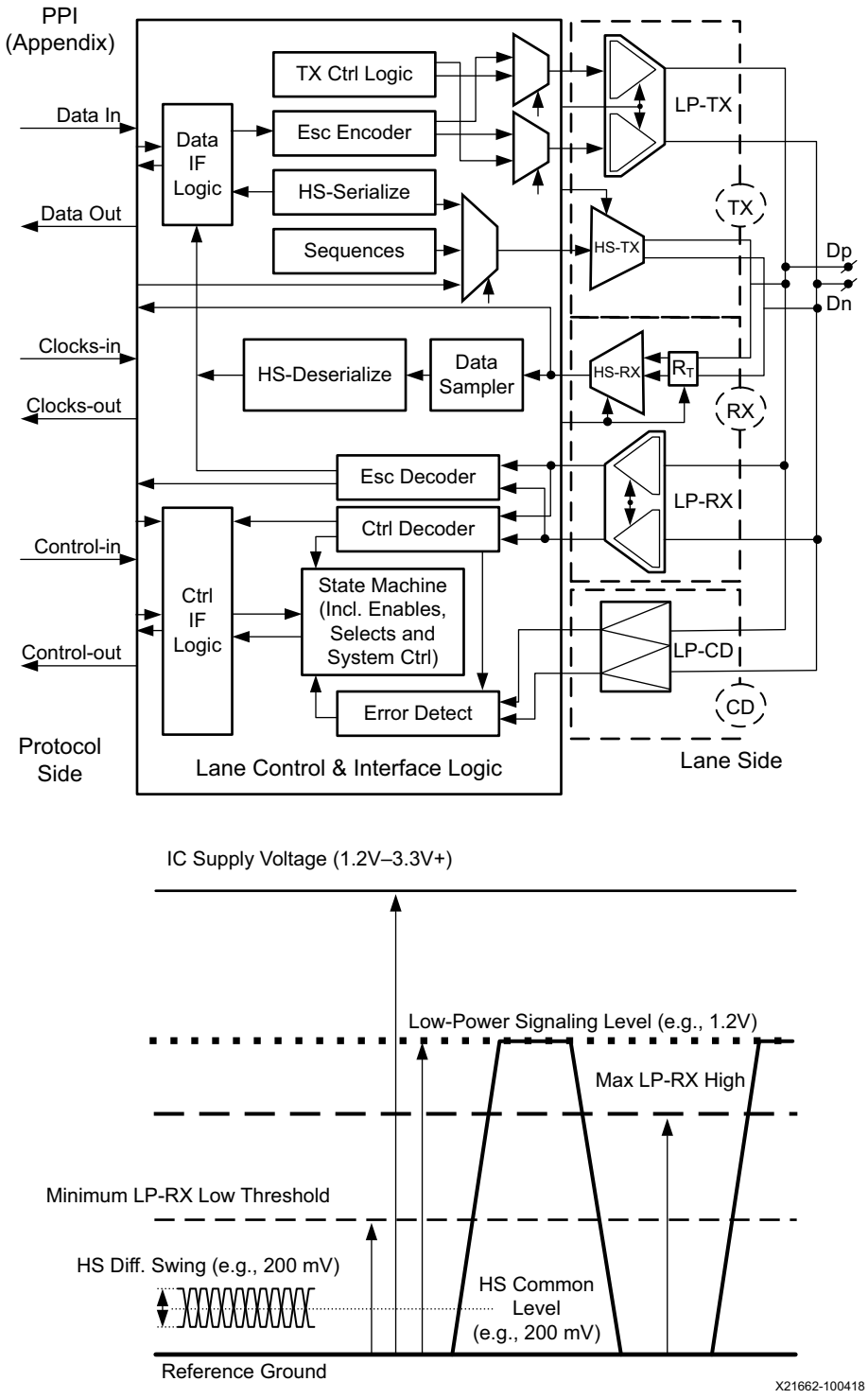


Figure 1: **D-PHY Overview**

The line rate that the MIPI interface can support is also gradually developing towards 2.5G or higher (such as Table 1) to support the 4K, 8K, and even higher resolution images. It is also difficult to achieve this standard using SelectIO technology. A more advanced technology is needed to meet the higher speed requirements of the MIPI interface.

Table 1: **D-PHY Version Integration and Downward Compatibility**

		RX D-PHY Specification Version									
		D-PHY v2.1		D-PHY v2.0		D-PHY v1.2		D-PHY v1.1		D-PHY v1.0	
		Max Speed (Gb/s)	Deskew Initialization	Max Speed (Gb/s)	Deskew Initialization	Max Speed (Gb/s)	Deskew Initialization	Max Speed (Gb/s)	Deskew Initialization	Max Speed (Gb/s)	Deskew Initialization
TX D-PHY Specification Version	D-PHY v1.0	1.0	–	1.0	–	1.0	–	1.0	–	1.0	–
	D-PHY v1.1	1.5	–	1.5	–	1.5	–	1.5	–	1.0	–
	D-PHY v1.2	2.5	Yes	2.5	Yes	2.5	Yes	1.5	–	1.0	–
		1.5	–	1.5	–	1.5	–				
	D-PHY v2.0	4.5	Yes	4.5	Yes	2.5	Yes	1.5	–	1.0	–
		1.5	–	1.5	–	1.5	–				
	D-PHY v2.1	4.5	Yes	4.5	Yes	2.5	Yes	1.5	–	1.0	–
		1.5	–	1.5	–	1.5	–				

Notes:
1. Cells containing dashes (–) indicate that deskew initialization is not required.

DSI and CSI

The DSI is a high-speed serial interface between a peripheral, such as an active-matrix display module, and a host processor. The DSI uses D-PHY as a physical communication layer. Information transfer between the host and a peripheral can consist of one or more serial data lanes and a clock lane.

Between transceiver sessions, the differential data/clock lane or lanes can switch to and from a low-power (LP) transceiver state. Interfaces should be in the idle state when they are not actively transmitting or receiving high-speed data. Figure 2 illustrates the basic structure of a high-speed transmission. A DSI interface can have 1, 2, 3, or 4 data lanes. Wider interfaces are organized as multiples of 1, 2, 3, or 4 lanes (for example, 8 data lanes can be generated as 1 × 8 lanes or 2 × 4 data lanes).

IMPORTANT: *Eight data lanes can only be generated as 2 × 4 data lanes in the 2.5G MIPI D-PHY solution. Data lane 0 is NOT bidirectional.*

Only data lane zero can support bidirectional data transfer via low-power data transmission.

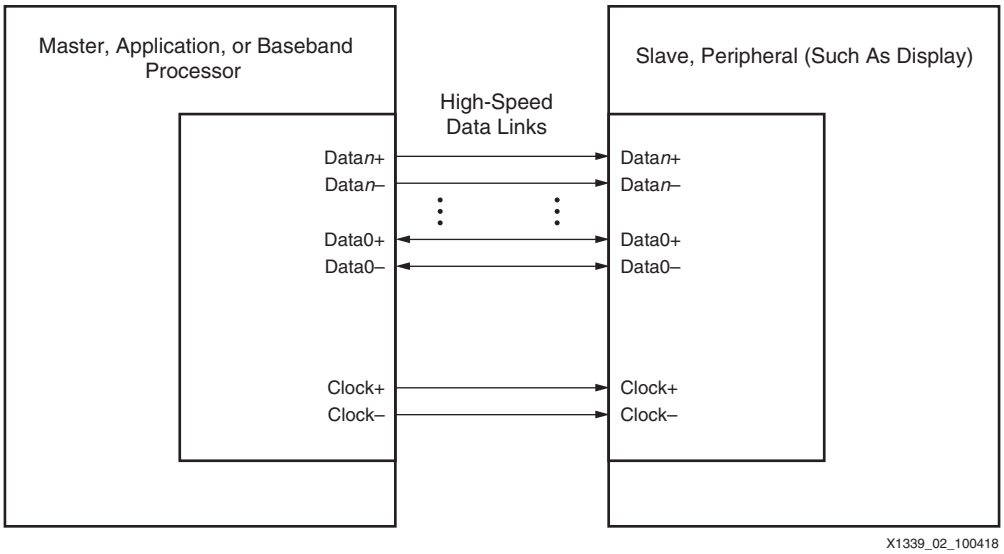


Figure 2: **Typical DSI Interface Structure**

All links between the host and the display peripheral are unidirectional, from host or master to display peripheral or slave. Only data lane zero can be bidirectional in the LP state.

The CSI is a high-speed serial interface between a peripheral, such as a camera, and a host processor. CSI also uses D-PHY as a physical layer interface as specified by the MIPI alliance. Figure 3 illustrates the connections between the CSI transmitter and the receiver interface. A CSI interface can have 1, 2, 3, or 4 data lanes. If more data lanes are necessary, the interface is organized as a multiple of 1, 2, 3, or 4 lanes (for example, 8 data lanes can be generated as 1 × 8 lanes or as 2 × 4 data lanes).

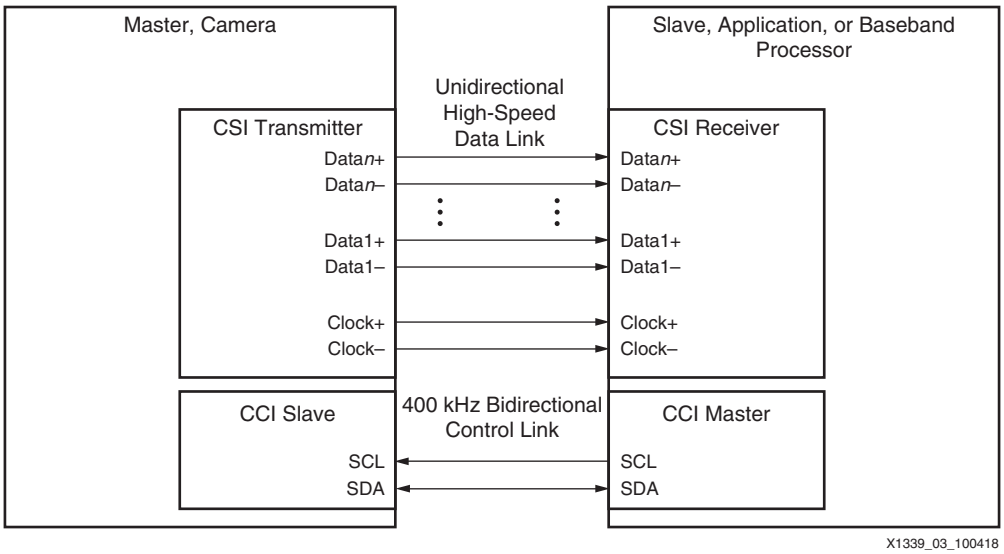


Figure 3: **Typical CSI Interface Structure**

The CSI transmission interface is composed of up to eight unidirectional differential serial lanes and a clock lane running in high-speed (HS) mode. Transmitters and receivers support continuous clock behavior and, optionally, non-continuous clock behavior. The control interface (referred to as CCI in Figure 3) is a bidirectional control interface and operates in the LP state.

Note: If the MIPI design does not support system synchronization clock mode between transmitters and receivers, only continuous clock mode can be supported for the 2.5G MIPI D-PHY solution. For more information, see D-PHY Equivalent.

D-PHY

Traditionally, interfaces between components on a printed circuit board (PCB) are based on single-ended parallel buses at low bit rates (LVCMOS), differential high-speed serial buses, or single differential channels.

The D-PHY provides an extension to this structure by turning the low-speed, low-power interface to the serial format of the high-speed differential interface, so that both are combined into a single serial interface. With this method, the D-PHY provides a flexible high-speed differential and low-speed, low-power single-ended serial interface solution for interconnection between components within one product.

The D-PHY specification is written with application specific standard products (ASSPs) or ASIC implementations in mind. It includes and combines scalable low-voltage signaling (SLVS) (high-speed) and LVCMOS (low-power) I/O into a single differential pair of wires (see Figure 1). As previously mentioned, FPGAs do not yet support native D-PHY-compliant I/O except in UltraScale+ FPGAs.

This application note presents a new method of supporting high-speed 2.5G MIPI on FPGAs with the simplest peripherals. The physical D-PHY specifications are listed in Table 2.

Table 2: **D-PHY Specifications**

Parameter	Value
Minimum number of data pins per direction	4
Minimum configuration	4 pins half duplex
Minimal UniPro configuration	8 pins
Medium	≤ 300 mm PCB, flex or micro coax
Data rate per lane:	2.5 Gb/s
Maximal HS rate	2.5 Gb/s
Minimal HS rate	≤ 80 Mb/s
LP rate	≤ 20 Mb/s
Electrical signaling:	
HS	SLVS-400
LP	LVCMOS-1.2V
HS clocking method	DDR source synchronous
HS line coding	None or 8B9B
Receiver clock data recovery (CDR) required	No
Suited for optical or repeater	No

100 MHz to 2.5G High-Speed D-PHY Emulation

This application note provides a solution for connecting an FPGA to a MIPI-compatible device. This is accomplished external to the FPGA I/O interface design to fully or partially emulate the D-PHY functionality. The decision to use fully or partially supported D-PHY specifications should be made based on cost, quantity, performance, and other design specific criteria. This application note only provides application possibilities.

Although the basic D-PHY emulation circuit can support bidirectional data and clock lanes, most customers and Xilinx have chosen to only support unidirectional functionality.

As shown in Figure 1, the D-PHY has two distinct functions:

- Lane control and interface logic (LCIL)
- Lane side logic (LSL)

This application note focuses on the PMA (PHY) side. Refer to the MIPI interface logic hierarchical design of *MIPI D-PHY v4.1 Product Guide* (PG202) [Ref 3] reproduced in Figure 4, providing the I/O circuit and lane side logic reference design of the PMA layer. It also includes the design methodology of these two parts. The physical coding sublayer (PCS) design is not included in this application note.

Note: The reference design has been tested with Xilinx partner Northwest Logic's CSI-2 Controller Core V2 and DSI-2 Controller Core. To test with another vendor's D-PHY PCS, contact Xilinx sales to reassess the solution.

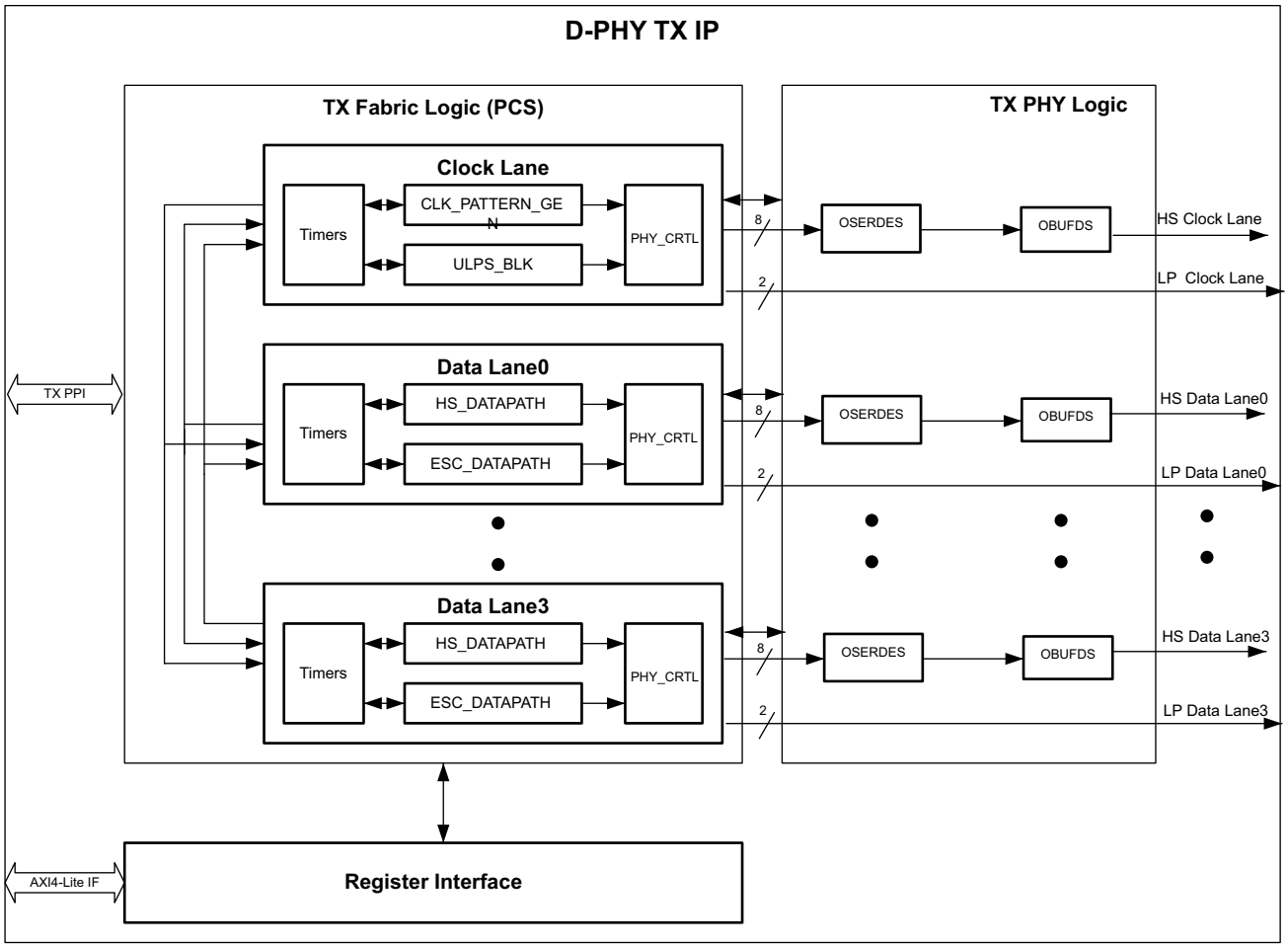


Figure 4: **MIPI D-PHY TX (Master) Core Architecture for 7 Series FPGA Family**

The following summarize the key features of the 2.5G D-PHY PMA:

- The FPGA I/O can support the time-sharing transmission of LP and HS signals on the same physical channel of the PCB, and ensure the integrity of the signal quality, conforming to the D-PHY electrical specifications.
- The design meets the timing requirements of the D-PHY specification and switches LP and HS signals correctly.

Before UltraScale+ FPGAs, the traditional D-PHY solution connected the HS signal to the differential standard of the FPGA, connected the LP signal to the LVCOMS I/O, and then adapted to the same physical channel by the resistor bridge on the PCB. At the same time, the PCS layer logic was based on the recognition of the protocol layer by analysis data, supported the signal being switched between the differential I/O and the LVCMOS, and realized the 800 Mb/s D-PHY interface standard (refer to *D-PHY Solutions* (XAPP894) [Ref 4]).

The UltraScale+ FPGA integrated the D-PHY buffer driver in the I/O block (IOB), which directly supports the LP and HS signals transmitted on a pair of SelectIO interfaces (refer to *MIPI D-PHY v4.1 Product Guide* (PG202) [Ref 3]).

The reference design accompanying this application note is similar to the one for *D-PHY Solutions* (XAPP894). However, it uses GTH transceivers instead of traditional differential I/O and also contains some advanced technology in the GTH transceivers to achieve a 2.5 Gb/s line rate high-speed D-PHY interface.

Note: The GTH transmitter can drive a 50Ω load referenced to GND with DC coupling.

The QPI sensors of the GTH receiver support the MIPI LP signal input directly. The equalizer of the GTH receiver supports up to a 6 dB stressed eye-diagram. The GTH receiver supports 5x oversampling to support a burst data stream in the MIPI. The CDR in the GTH receiver can be controlled manually. Thus, a manual CDR can be created to simplify the design for the asynchronous clock mode of the MIPI interfaces at reduced cost.

The following is a summary of the design methodology:

- The circuit design meets electrical characteristics and ensures signal integrity:
  - TX side: High-speed field effect transistor (FET) switches replace resistor networks to connect GTH (HS) and SelectIO (LP) signals to the same PCB channel (see Figure 10).
  - RX side: GTH transceiver with embedded Quick Path Interconnect (QPI) unit real-time detection line level allows for automatic switching between LP and HS data channel, and direct simulation of D-PHY driver behavior connecting to the standard MIPI interface. No external circuit is needed (see Figure 11).

  The technology in the GTH transceiver guarantees the receiving and transmitting ability of the HS signal in the 2.5G range.

- Lane side logic design - HS and LP signal switching:
  - TX side: The PCS layer logic determines the transmission state of the transmitted data according to the D-PHY protocol and sends out the channel selection signal (SEL). The HS and LP signal switching is realized through the PMA layer through the SEL signal to the SEL pin of the FET switch.

  Note: Because the data channel and the clock channel LP and HS switch timing is not exactly the same, and the CLK channel cannot be interrupted in continuous clock mode, the clock channel and the data channel need an independent FET switch. Two separate SEL signals are sent out from the MIPI control logic.

  - RX side: Based on the difference of HS and LP level standards, the transceiver QPI unit detects the line level status in real time and automatically sends data to the LP and HS data channel within the transceiver. The PMA layer transmits an LP signal to the PC layer. The PCS layer needs to monitor the LP channel state to generate a data mask signal so that the output HS signal from the PMA layer is collected correctly in the PCS layer.

  Note: On the FPGA side, the MIPI CLK channel is transmitted directly through the GTH transceiver without connecting to a separate FPGA I/O clock pin.

  When the receiving channel switches to the HS signal, the data received by the internal LP data channel is always 0. More information can be seen in the logical design section in D-PHY Equivalent.

As shown in Figure 5, according to the D-PHY interface timing standard, the PCS layer on the sending side can indicate FET-switch switching by analyzing the $T_{HS-SETTLE}$ and $T_{HS-TRAIL}$ status. Considering the timing requirements of the data channel and clock channel, there is a limit for FET switching time. For more information, see TX Circuit Design Guidance.

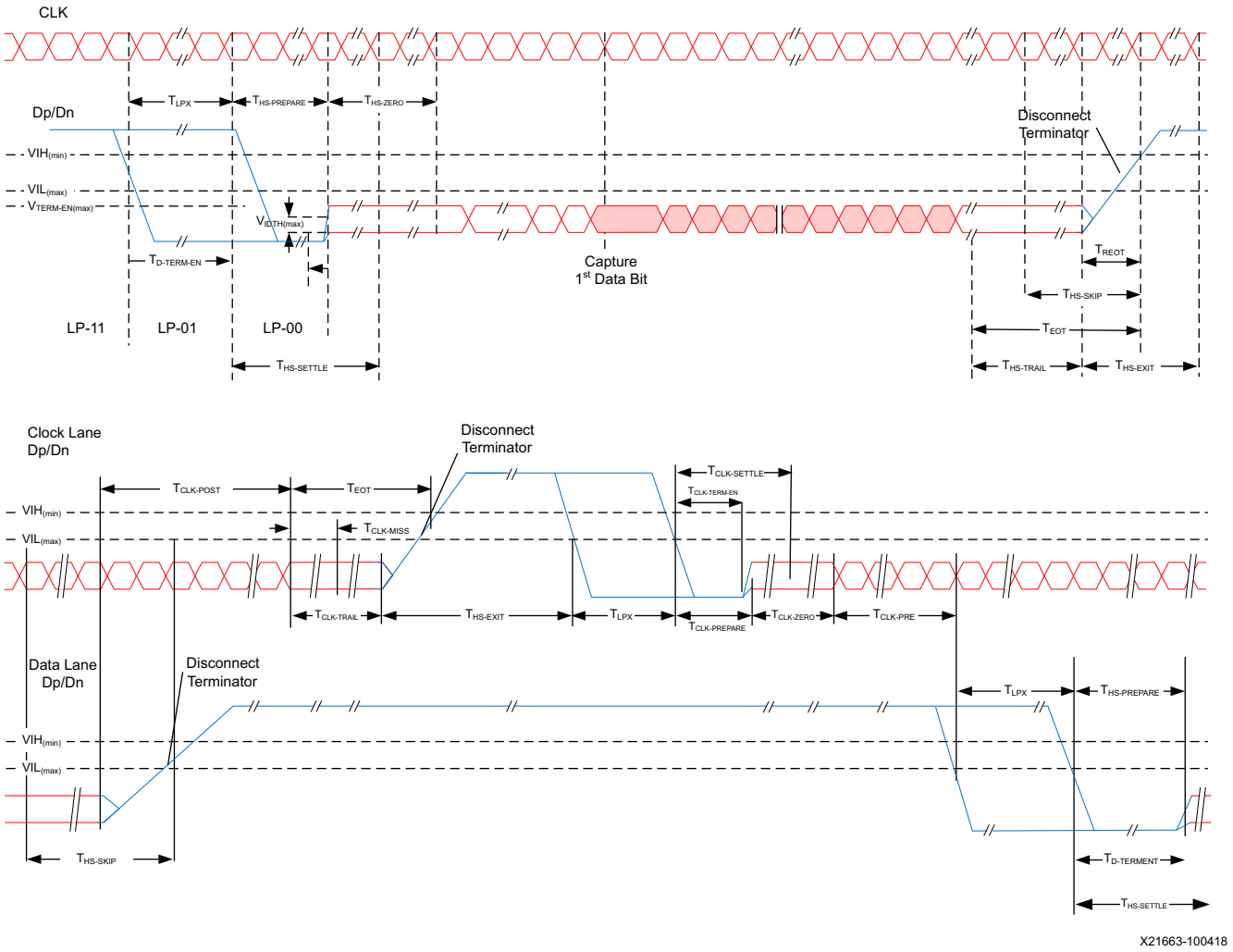


Figure 5: **High-Speed Data Transmission in Bursts/Switching the Clock Lane between Transmission and Low-Power Mode**

High-Speed Interface

The differential high-speed part of the D-PHY can be mimicked by an FPGA using differential high-speed I/O such as LVDS and DIFF_HSTL or differential HSTL. In all cases, the I/O levels of the FPGA need to be adapted to the low-swing, SLVS style I/O specified for the D-PHY.

D-PHY High-Speed I/O Specifications

Table 3 shows the D-PHY high-speed I/O specifications stipulated in the *MIPI Alliance specification for D-PHY*. These meet the specifications for the SLVS standard in the JESD 8-13 SLVS specification [Ref 5].

When an FPGA is used to mimic a D-PHY, the differential FPGA standards used must meet these SLVS specifications. This section highlights the differential FPGA standards that can be used, with some external signal shaping components, as a D-PHY compliant solution.

Table 3: **D-PHY HS Transmitter DC Specifications**

Parameter	Description	Minimum	Nominal	Maximum	Units
Transmitter					
EQTX1(1)	De-emphasis option 1	2.5	3.5	4.5	DB
EQTX2(1)	De-emphasis option 2	6	7	8	DB
VCMTX(2)	HS transmit static common mode voltage	150	200	250	mV
VCMTX_HalfSwing(2)(4)	HS transmit static common-mode voltage in half swing mode	75	100	250	mV
|ΔVCMTX(1,0)|(3)	VCMTX mismatch when output is differential-1 or differential-0			5	mV
|VOD|(2)	HS transmit differential voltage	140	200	270	mV
|VOD_HalfSwing|(2)(4)	HS transmit differential voltage in half-swing mode	70	100	135	mV
|ΔVOD|(3)	VOD mismatch when output is differential-1 or differential-0			14	mV
VOHHS(2)	HS output high voltage			360	mV
ZOS	Single-ended output impedance	40	50	62.5	
ΔZOS	Single-ended output impedance mismatch			20	%

Notes:

1. When the supported data rate is greater than 2.5 Gb/s, conformance requirements for the transmitter are defined through the eye diagram. The values for equalization in this table are informative.
2. This is the value when driving into load impedance anywhere in the ZID range.
3. A transmitter should minimize ΔVOD and ΔVCMTX(1,0) to minimize radiation and optimize signal integrity.
4. Half swing mode is optional. It is an additional capability that a transmitter can support for better system power optimization.

Table 4: **D-PHY HS Transmitter AC Specifications**

Parameter	Description	Minimum	Nominal	Maximum	Units	Notes
ΔVCMTX (HF)	Common-level variations above 450 MHz			15	mV_{RMS}	
ΔVCMTX (LF)	Common-level variation between 50–450 MHz			25	mV_{PEAK}	

Table 4: **D-PHY HS Transmitter AC Specifications *(Cont'd)***

Parameter	Description	Minimum	Nominal	Maximum	Units	Notes
t_R and t_F	20%–80% rise time and fall time			0.3	UI	(1)(2)
				0.35	UI	(1)(3)
		100			ps	(4)

Notes:

1. UI is equal to 1/(2 × fh), where fh is the fundamental frequency of the operating data rate.
2. This is applicable when supporting maximum HS bit rates ≤ 1 Gb/s (UI ≥ 1 ns).
3. This is applicable when supporting maximum HS bit rates > 1 Gb/s (UI ≤ 1 ns) but ≤ 1.5 Gb/s (UI ≥ 0.667 ns).
4. This is applicable when supporting maximum HS bit rates ≤ 1.5 Gb/s. However, to avoid excessive radiation, bit rates < 1 Gb/s (UI ≥ 1 ns) should not use values below 150 ps.

Table 5: **D-PHY HS Receiver DC Specifications**

Parameter	Description	Minimum	Nominal	Maximum	Units
VCMRX (DC)(1)(2)	Common mode voltage HS receive mode	70		330	mV
ZID	Differential input impedance	80	100	125(3)	Ω
ZID_OPEN	Differential input impedance in unterminated mode(4)	10K			Ω

Notes:

1. This excludes possible additional RF interference of 100 mV peak sine wave beyond 450 MHz.
2. This table value includes a ground difference of 50 mV between the transmitter and the receiver, the static common-mode level tolerance, and variations below 450 MHz.
3. ZID can be higher than 125Ω in unterminated mode.
4. Unterminated mode for HS-RX is optional. This mode can only be used when a transmitter is in half swing mode. ZID_OPEN is defined for a differential voltage with maximum amplitude of |VOD_HalfSwing| and within the common voltage range of VCMTX_HalfSwing.

Table 6: **D-PHY HS Receiver AC Specifications**

Parameter	Description	Minimum	Nominal	Maximum	Units	Notes
ΔVCMRX (HF)	Common-mode interference beyond 450 MHz			100	mV	(2)(5)
				50	mV	(2)(6)
ΔVCMRX (LF)	Common-mode interference 50–450 MHz	–50		50	mV	(1)(4)(5)
		–25		25	mV	(1)(4)(6)
VIDTH	Differential input high threshold			70	mV	(5)
				40	mV	(6)
VIDTL	Differential input low threshold	–70			mV	(5)
		–40			mV	(6)
VIHHS	Single-ended input high voltage			460	mV	(7)
VILHS	Single-ended input low voltage	–40			mV	(7)
ZTERM-EN	Single-ended threshold for HS termination enable			450	mV	

Table 6: **D-PHY HS Receiver AC Specifications *(Cont'd)***

Parameter	Description	Minimum	Nominal	Maximum	Units	Notes
Ccm	Common-mode termination			60	pF	(3)

Notes:

1. This excludes "static" ground shift of 50 mV.
2. ΔVCMRX(HF) is the peak amplitude of a sine wave superimposed on the receiver inputs.
3. For higher bit rates, a 14 pF capacitor is needed to meet the common-mode return loss specification.
4. The voltage difference is compared to the DC average common-mode potential.
5. This is for devices supporting data rates ≤ 1.5 Gb/s.
6. This is for devices supporting data rates > 1.5 Gb/s.
7. This excludes possible additional RF interference of 100 mV peak sine wave beyond 450 MHz.

VCMTX, |VOD| and VCMRX (DC) are important indicators to pay close attention to. Some special techniques must be used to meet the requirements of these indicators.

GTH Transceivers

From the GTH transceiver DC specifications table in *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS892) [Ref 6], $V_{CMOUTDC} = V_{MGTAVTT}/2 - D_{VPPOUT}/4$. The specified level for LP single-ended I/O with D-PHY is 1.2V. If D_{VPPOUT} = 1000 mV, $V_{CMOUTDC}$ = 1200/2 – 1000/4 = 350 mV. This is still larger than the D-PHY requirement of the maximum V_{CMTX} being 250 mV. Fortunately, D_{VPPOUT} is far greater than the |VOD| requirement, which makes it possible to insert attenuators.

Low-Power Interface

The specified level for LP single-ended I/O with D-PHY is 1.2V. When a 100 Mb/s–2.5 Gb/s D-PHY equivalent circuit is implemented in the FPGA, the LP signal on the transmitting side is sent through the SelectIO interface in the FPGA. The GTH transceiver on the receiving side can be compatible with receiving HS and LP signals simultaneously. The HSUL_12_DCI level standard or other 1.2V single-ended level standards with series resistors are recommended for sending side LP signals because of better signal integrity.

IMPORTANT: *LP signals must be placed in a 1.2V I/O bank.*

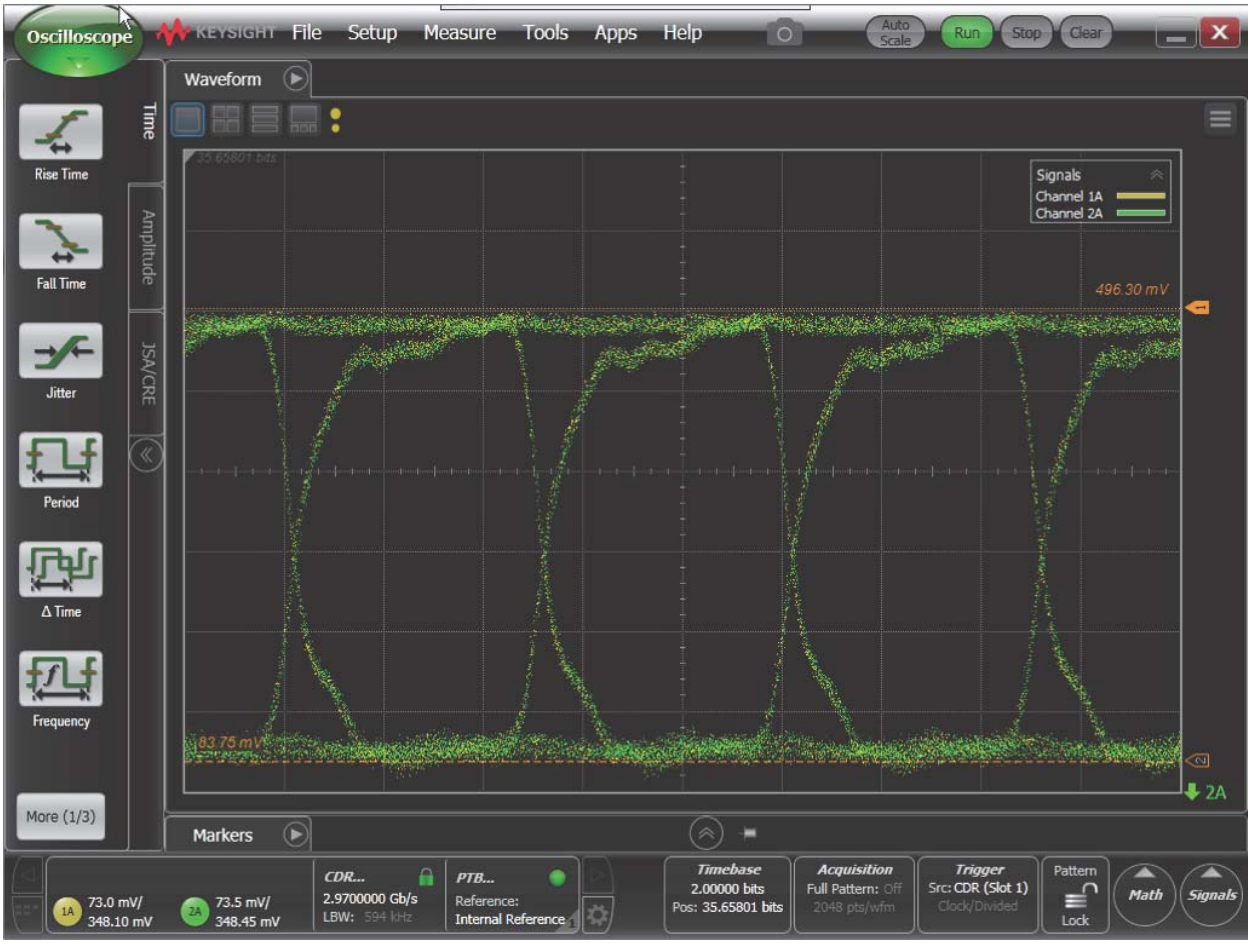
D-PHY Equivalent

GTH Transceiver Solution

To meet the requirements of the D-PHY HS electrical index, the QPI module on the TX side of the GTH transceiver can be DC-coupled to drive a ground load with an output lower common-mode voltage. The RX side can receive a low common-mode voltage signal with DC coupling and also have a 0.6V voltage detection sensor.

Hardware Feature

The waveform in Figure 6 shows the signal status of a GTH transceiver driving 50Ω to ground a load. The high level is at 496.3 mV, and the low level is at 83.75 mV. Thus, the common-mode voltage is 290.0 mV, which is close to 200 mV of the D-PHY TX. The only requirement is to design an approximate –3 dB divider attenuator to meet the requirements. Fortunately, the output of the GTH transceiver is 412.55 mV, which is also larger than the 140 mV amplitude requirement of the D-PHY TX. The right conditions have been created for the insertion of the attenuator. Thus, the design was modeled and simulated in the Keysight ADS environment.



X21688-100418

Figure 6: **Output Waveform of GTH Transceiver Driving 50Ω Load**

Simulation

As shown in Figure 7, a current mode logic (CML) circuit model was first built to drive a 50Ω-to-ground load so that its output approximates the actual GTH transceiver test data (Figure 8).

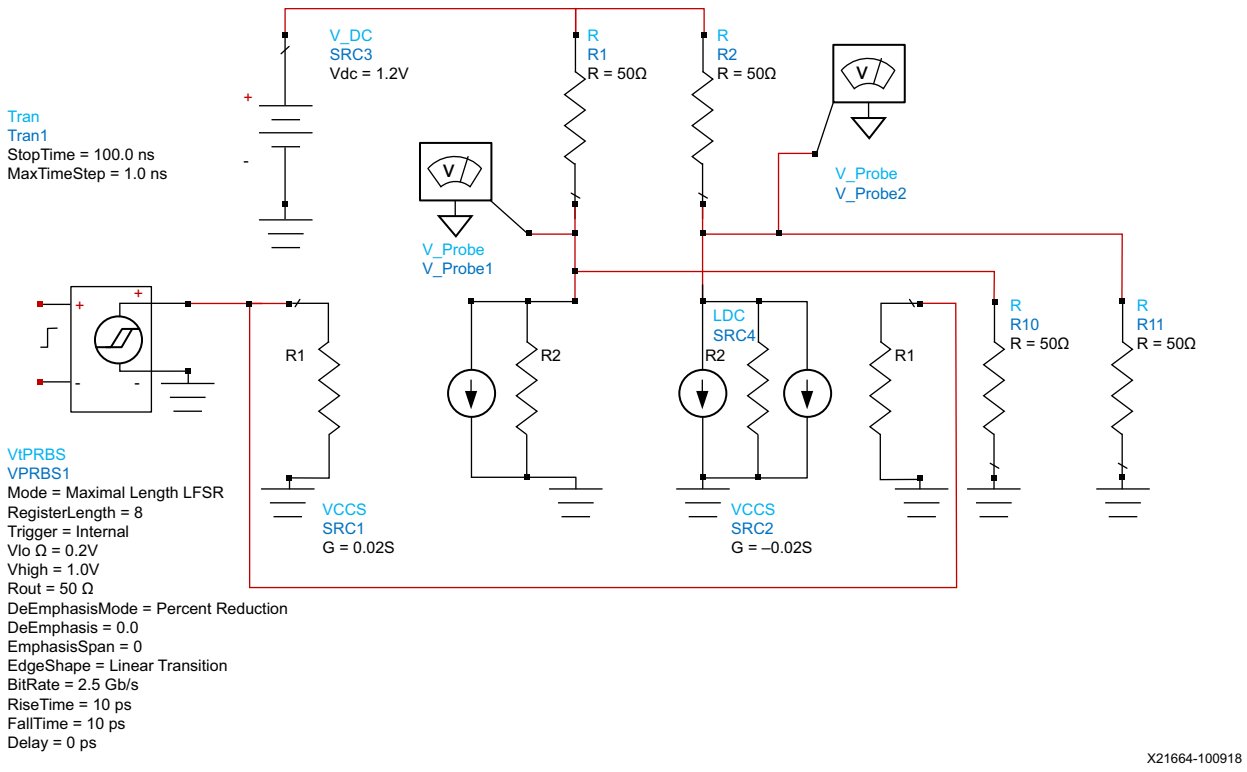


Figure 7: **ADS Modeling for GTH Transceiver Driving 50Ω Load**

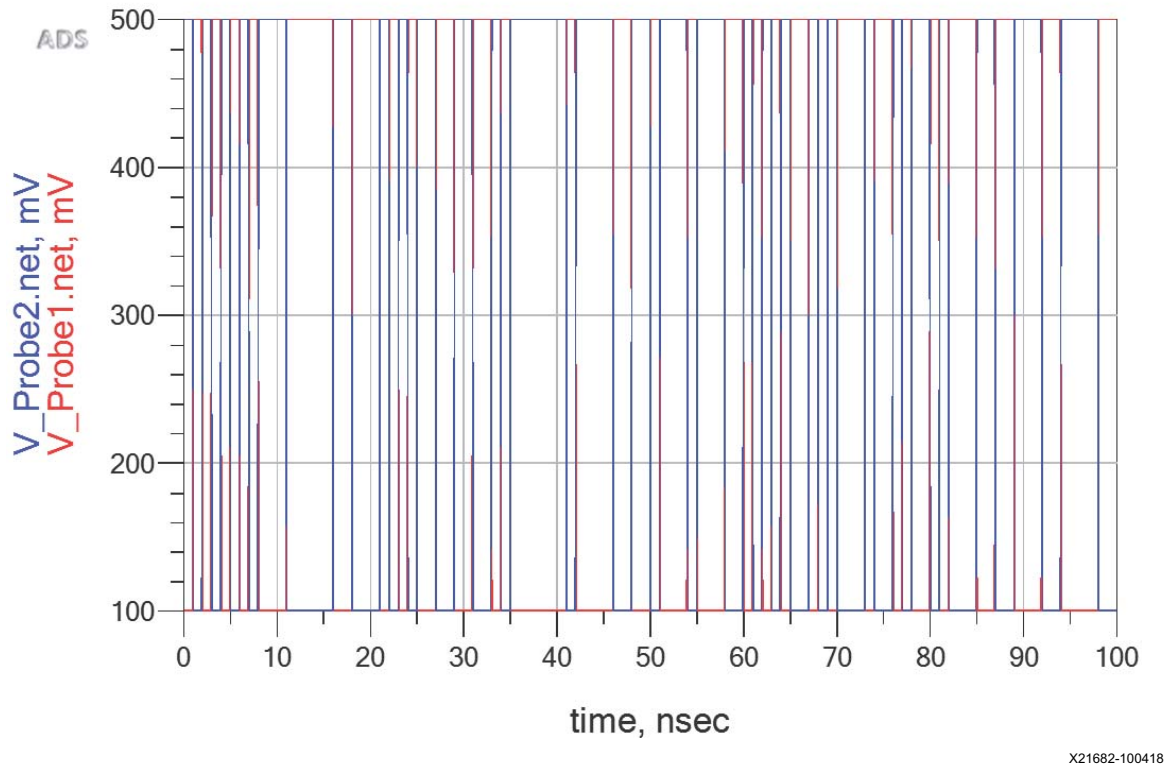


Figure 8: **Output Waveform of ADS Modeling for GTH Transceiver Driving 50Ω Load**

The 50Ω-to-ground load is then replaced with a T-type attenuator of about 5 dB to observe the output common-mode voltage and differential swing. As shown in Figure 9, the common-mode voltage and differential swing can meet the requirements of the HS TX side of the D-PHY.

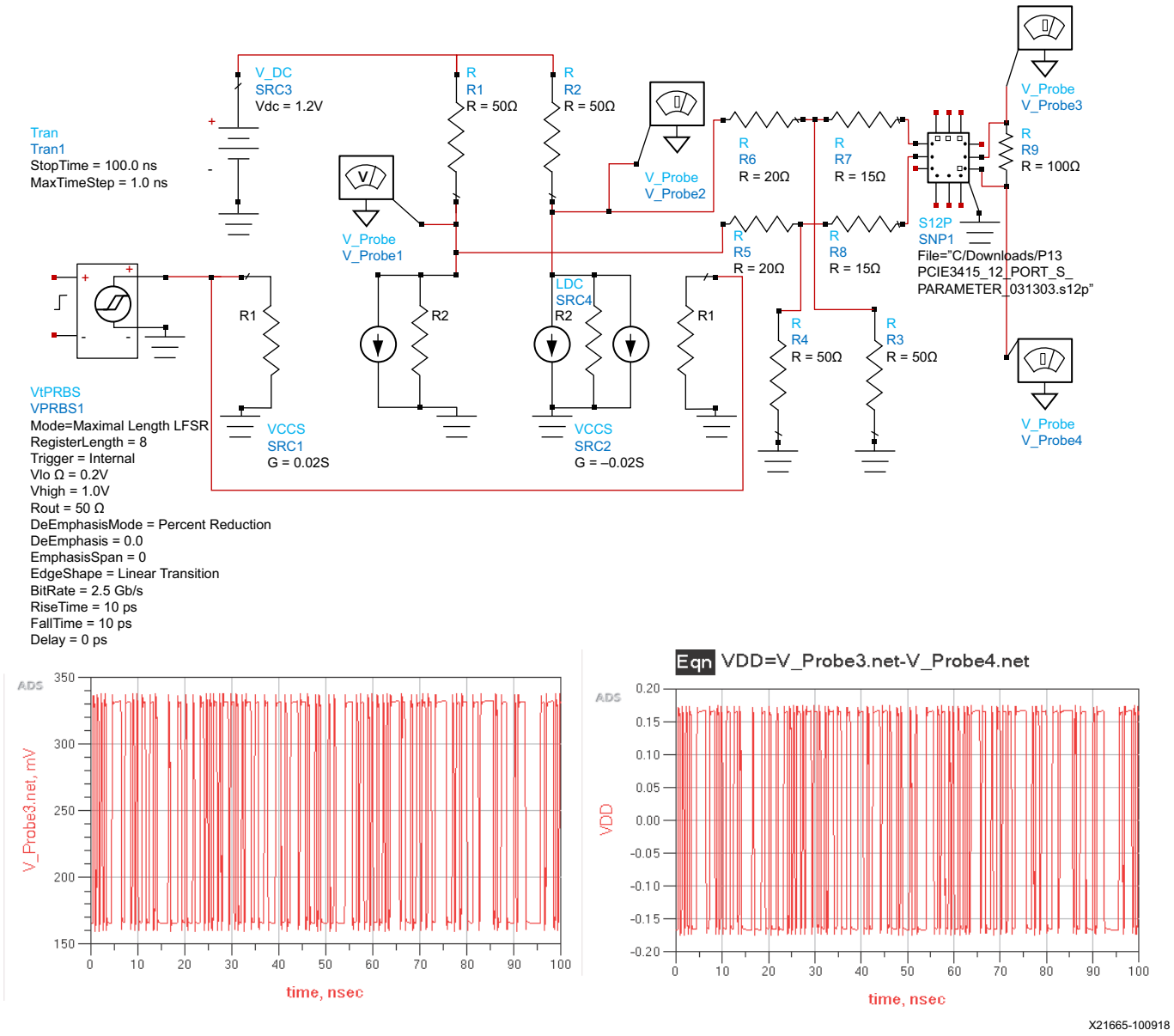


Figure 9: **Output Single-End and Differential Waveform after the Attenuator**

TX Circuit Design Guidance

Based on the above simulation results, external high-speed FET analog switches are used to construct a D-PHY TX circuit, as shown in Figure 10.

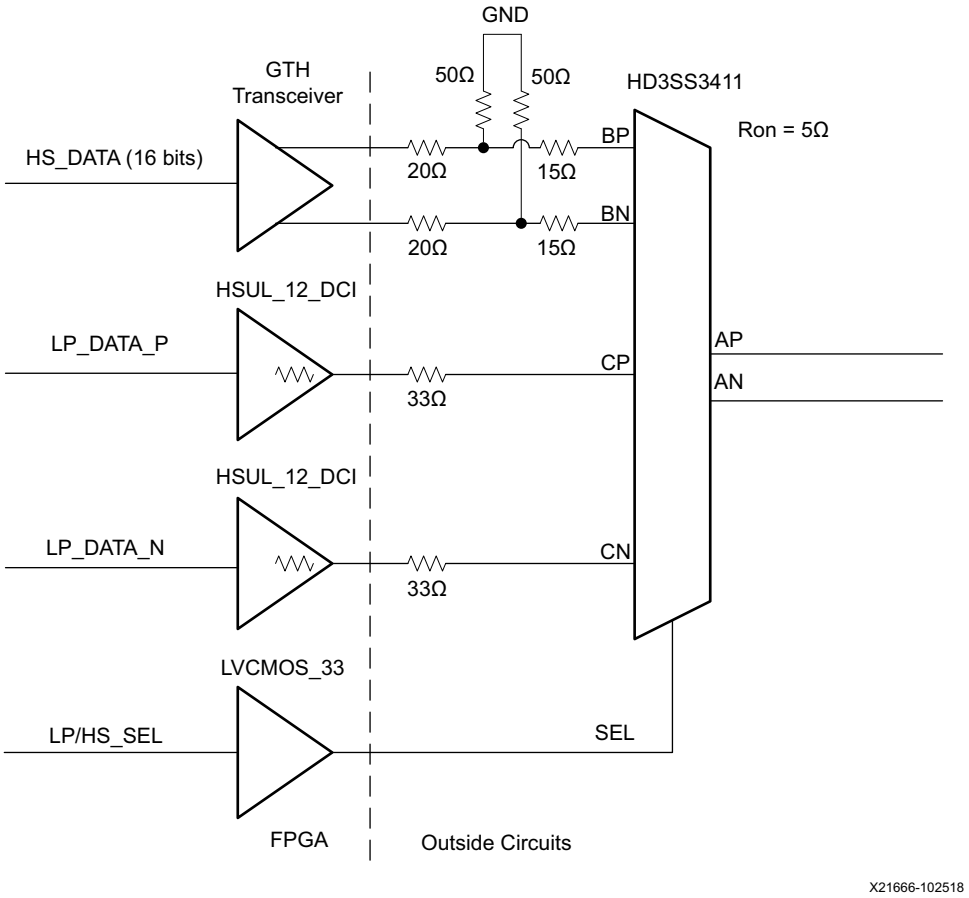


Figure 10: **D-PHY TX Circuit**

Based on the above circuit, the following points should be considered when selecting high-speed FET analog switches:

- The conduction resistance R_{ON} must be less than 20Ω.
- Switch delay T_{SW} must be less than $T_{HS\text{-}SETTLE}$ and $T_{HS\text{-}EXIT}$ (100 ns).
- Near-end crosstalk should be less than –30 dB.
- The –1 dB bandwidth is greater than 1.25 GHz.

RX Circuit Design Guidance

As shown in Figure 11, the RX can completely support the D-PHY's receiving electrical requirements without any external devices compared to the TX. When enabling RXQPIEN, RXQPISENP/N can realize LP signal detection.

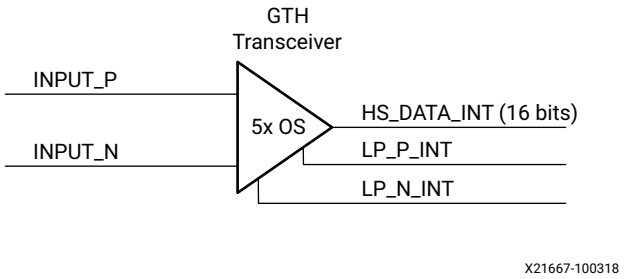


Figure 11: **D-PHY RX Circuit in GTH Transceiver**

End-matching in the GTH transceiver is not disconnected in LP mode like in the real D-PHY, so LP signals are attenuated by end-matching. Figure 12 shows that in this case, the low level is about 0.4V, the high level is about 0.8V, and both can still be reliably determined by RXQPISENP/N.

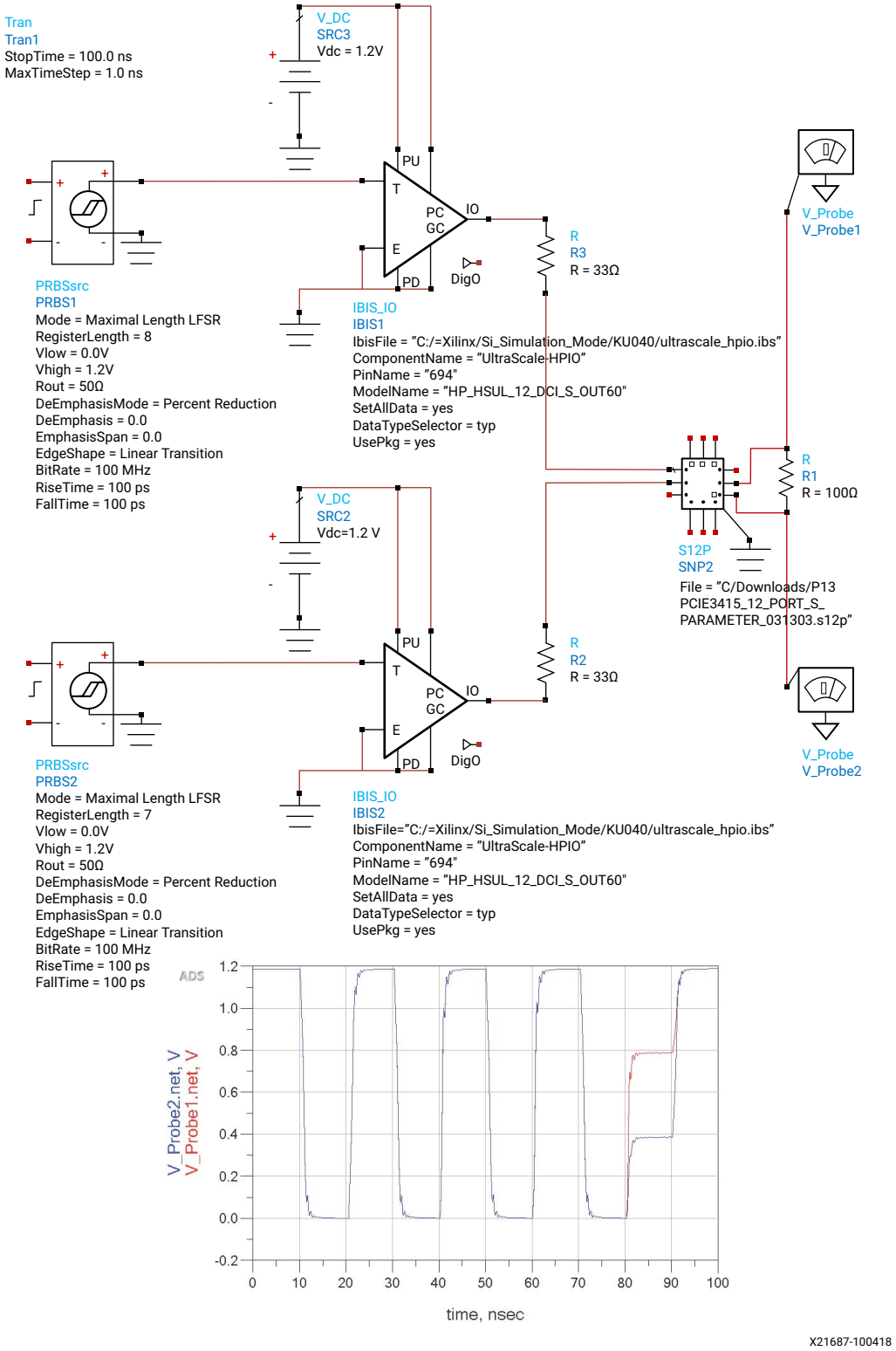


Figure 12: **ADS Modeling for LP Driving 100Ω Load and Output Waveform**

IMPORTANT: *Because the RX works in DC coupling mode, and VICM_DC is around 200 mV, the RX must be set as follows:*

· *RX_CM_SEL[1:0] = 2'b10*

· *RXDFEAGCCTRL[1:0] = 2'b10*

These two parameters are independently configured for each transceiver receiving channel, which is already set in the GTH Wizard in the example project. The example is a 4+1 lane default design. If this number of MIPI channels is not desired, these two parameters should be modified when the GTH channels are used for other purposes.

When the HS signal is burst, the GTH transceiver must work in oversampling mode using logic to achieve burst reception. To simplify the design, the system should synchronized to the clock. If the transmission-side reference clock cannot be provided to the receiver, the clock channel is required to operate in continuous mode. The D-PHY RX CLK channel works in CDR mode. At the same time, the CDR phase parameters of the clock channel are copied to the other data channels in real time through the dynamic reconfiguration port (DRP).[1] In this way, the DATA channel can achieve 5 times oversampling precisely, thus simplifying the design. Moreover, PCS logic is used to calibrate the HS effective time window to further ensure the stability of phase processing. (For the details of the receiver clock system, refer to I/O Assign/Timing Constraints of the 2.5G MIPI Solution.)

Note: Often the TX and RX sides are in different devices (e.g., TX = camera, RX = FPGA). Non-continuous clock mode is NOT supported unless both sides are in the same FPGA and driven by the same reference clock.

Reference Clock

Depending on the line rate, different reference clock frequencies are needed. Refer to GTH Transceiver Reference Clock Selection and Calculation.

1. This technology is termed manual CDR-M-CDR.

Logic Feature/Timing

Figure 13 shows the structure of the 2.5 Gb/s transceiver D-PHY reference design. In this block diagram, Xilinx provides only the USER TOP (D-PHY PMA) reference design. The complete demonstration reference design needs to connect with Northwest Logic's MIPI IP core.

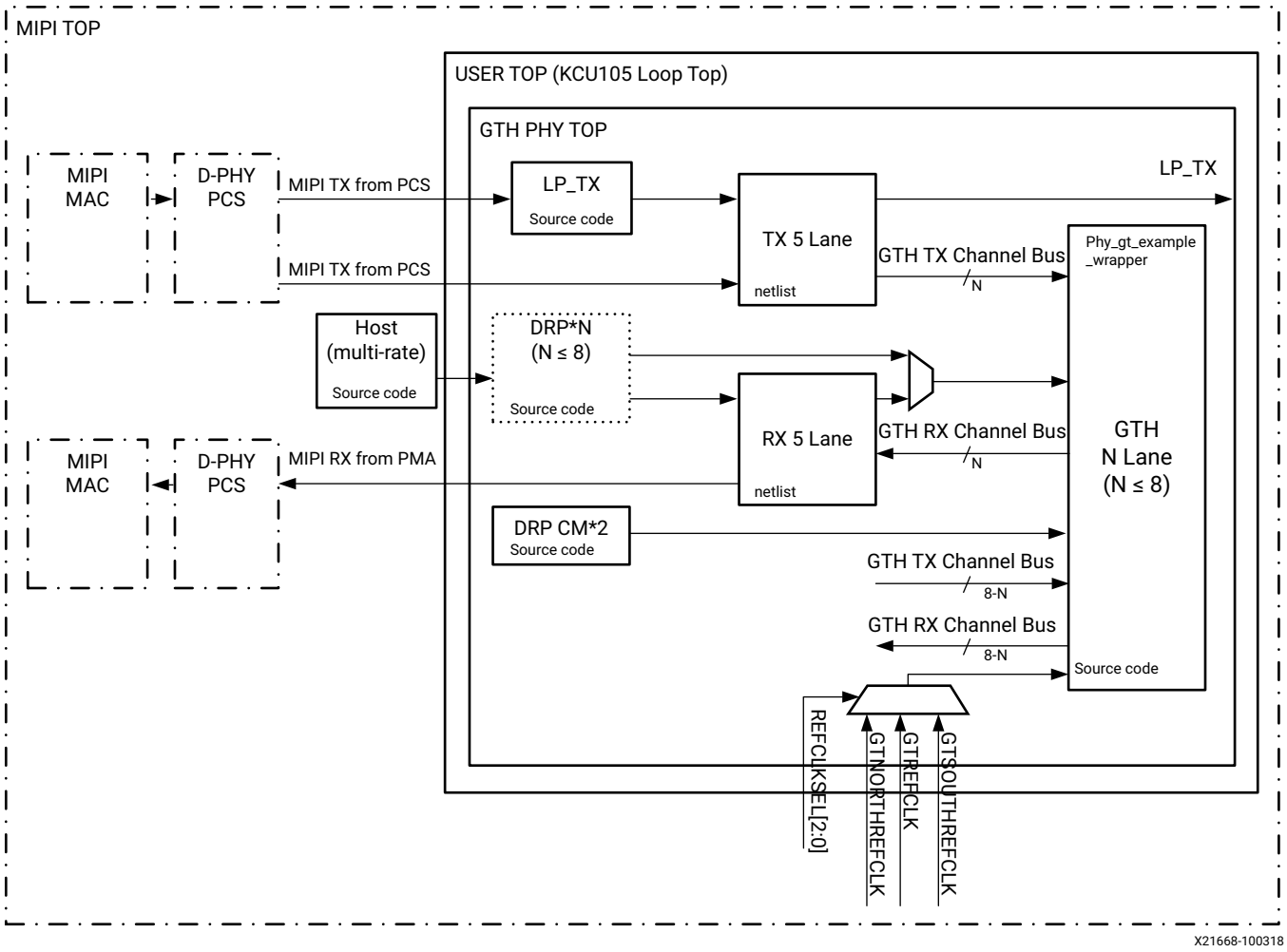


Figure 13: **2.5G MIPI D-PHY IP Core Architecture**

Note: The 2.5G D-PHY PCS and MIPI MAC IP core of the example design are provided by Xilinx IP partner Northwest Logic. For more information about the 2.5G D-PHY PCS and MIPI MAC IP core, refer to https://nwlogic.com.

The HDL hierarchy in the Vivado® tools project is shown in Figure 14.

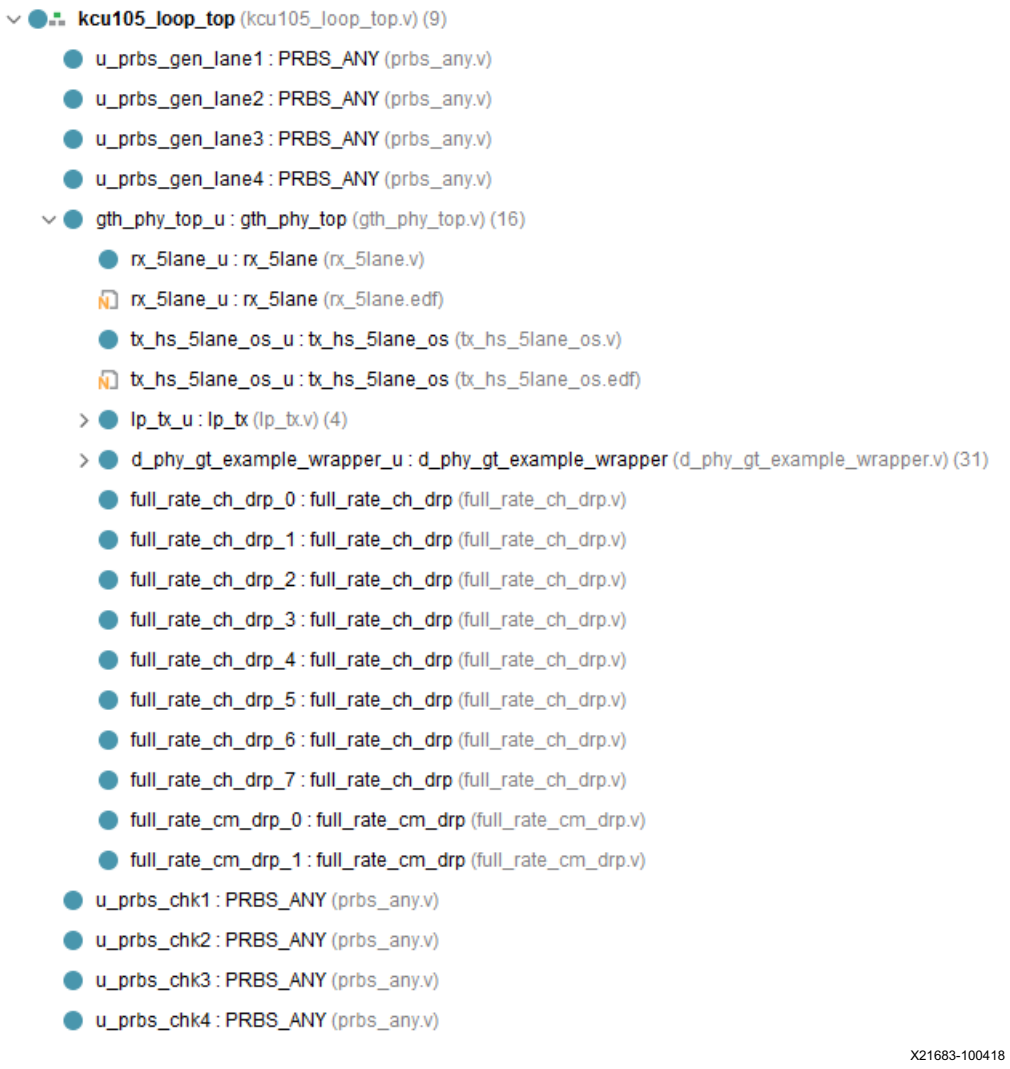


Figure 14: **HDL Hierarchy in Reference Design**

The following considerations must be made when dividing the logical structure of the design:

1. The design is independent of the D-PHY PMA layer and the D-PHY PCS layer at the logical level. Customers can design their own PCS layer or use the Xilinx third-party IP core.
2. USER TOP contains all transfer parameters required by the 2.5 Gb/s transceiver D-PHY module. At the same time, the physical location of the GTH transceiver reference clock is also processed. According to the GTH location on the PCB, the GTH transceiver reference clock is configurable. Customers need to modify the relevant code in `User_top.v` (`KCU105_loop_top.v/ZCU102_loop_top.v`) as well as the GTH transceiver reference clock pin location in the Xilinx design constraints (XDC). Refer to the Reference Clock Selection and Distribution section in *UltraScale Architecture GTH Transceivers User Guide (*UG576) [Ref 2] and 2.5 Gb/s GTH D-PHY Custom Design and GTH Transceiver Migration Guidelines.
3. Four modules are contained in `GTH_PHY_Top.v`: Phy_gt_example_wrapper (GTH), RX 5lane and TX 5lane (D-PHY PMA), and DRP module. The Phy_gt_example_wrapper (GTH) and DRP modules provide open source code that allow customers to customize the reference design.

4. The Phy_gt_example_wrapper (GTH) module is modified based on the standard <GTH transceiver example wrapper design> to meet the design requirements of the MIPI interface. The reference design contains the GTH transceiver interface of eight channels (in 2X GTH Quad).

   This module contains a standard GTH IP XCI file. The top module of the design takes out all ports of each transceiver channel. Through a logic channel adaptation layer, the transceiver channel used by the MIPI is connected to the TX/RX 5lane module from the GTH transceiver. Other GTH transceiver channel interfaces can be drawn out by the customer for other protocol transceiver interfaces.

   Customers can generate a GTH module through a GTH wrapper, customize the number of GTH channels and GTH location, and then connect to the RX/TX 5lane module (see Implementing the 2.5G Transceiver D-PHY Reference Design). For more information on the user-defined design methodology, refer to 2.5 Gb/s GTH D-PHY Custom Design and GTH Transceiver Migration Guidelines.

5. The RX/TX 5lane module processes MIPI data and clocks (HS and LP) transmitted over the transceiver channel. It also connects to the transceiver module through a logic channel adaptation layer to implement the D-PHY PMA layer function, providing the correct data for the PCS.

6. The DRP module is used to dynamically configure the GTH transceiver to support a multi-line rate MIPI interface. When the MIPI interface is in fixed-line rate mode, the DRP module is optional.

   Users can customize and modify the number and location of GTH channels, and then modify the DRP CH module enabling parameters: MIPI_TX_USED/MIPI_RX_USED (GTH_PHY_TOP.v).

7. The Host module (multi_rate_tb.v) is included in the simulation file that is used to control the DRP to change the MIPI line rate in real time. At the same time, the Host module is also used to select the clock source of the GTH channel between the QPLL and CPLL. The behavior of the Host module can be implemented in the CPU, or in the FPGA by HDL code. More details of this can be seen in Multi-Line Speed Rate Support (From 100 Mb/s to 2.5 Gb/s).

8. MIPI TX data LP and HS signals are sent to external FET switches independently from the FPGA I/O or GTH transceiver. The LP_TX module is used to send MIPI LP TX signals from the SelectIO interface.

Note: The reference design does not support CPLL mode when run on UltraScale+ FPGAs. For more information, see Porting the Design from UltraScale to UltraScale+ FPGAs.

IMPORTANT: *The mipi.xdc constraints file is a part of the design. It must be included in the target design.*

RX

The RX PHY core logic in Figure 15 helps to extract the HS data/clock and LP data/clock from the GTH transceiver. Because a bind transceiver (TX/RX) channel code is always generated when the GTH IP is generated, `GTH_PHY_TOP.v` helps to separate GTH RX and TX ports with the `Phy_gt_example_wrapper.v` file, and facilitates independent design of the RX 5lane and TX 5lane modules.

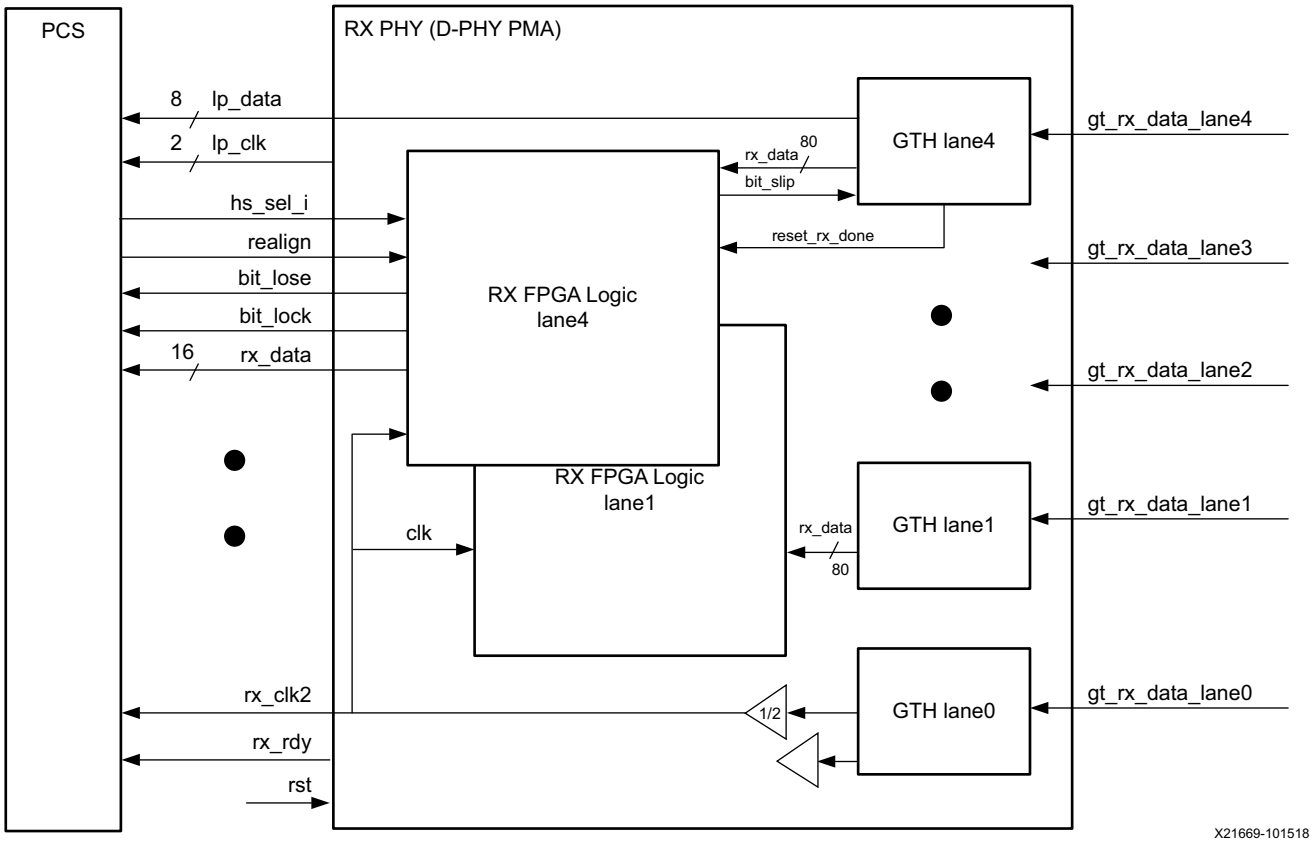


Figure 15: **2.5G MIPI D-PHY RX Logic Block**

LP Channel

All the LP data and clocks come from the RX_5lane module. These are asynchronous signals and constitute the first information to be used for the MIPI protocol handle because the HS data arrives later, based on the GTH structure.

HS Selection

The hs_sel_i signal comes from the MIPI controller, and any HS data flow should work when hs_sel is High. Otherwise, a bit loss error can occur. The hs_sel_i signal can work independently for each lane. It is generated based on the MIPI protocol, and it can be sourced by LP information. This signal timing is critical. Avoid having the HS data flow fall into the undefined I/O level range for the GTH transceiver. The hs_sel_i signal should start after LP-00 and end before $T_{HS\text{-}EXIT}$. Xilinx suggests keeping a margin of two rx_clk2 clock cycles in $T_{HS\text{-}ZERO}$ and $T_{HS\text{-}TRAIL}$ (see Figure 16).

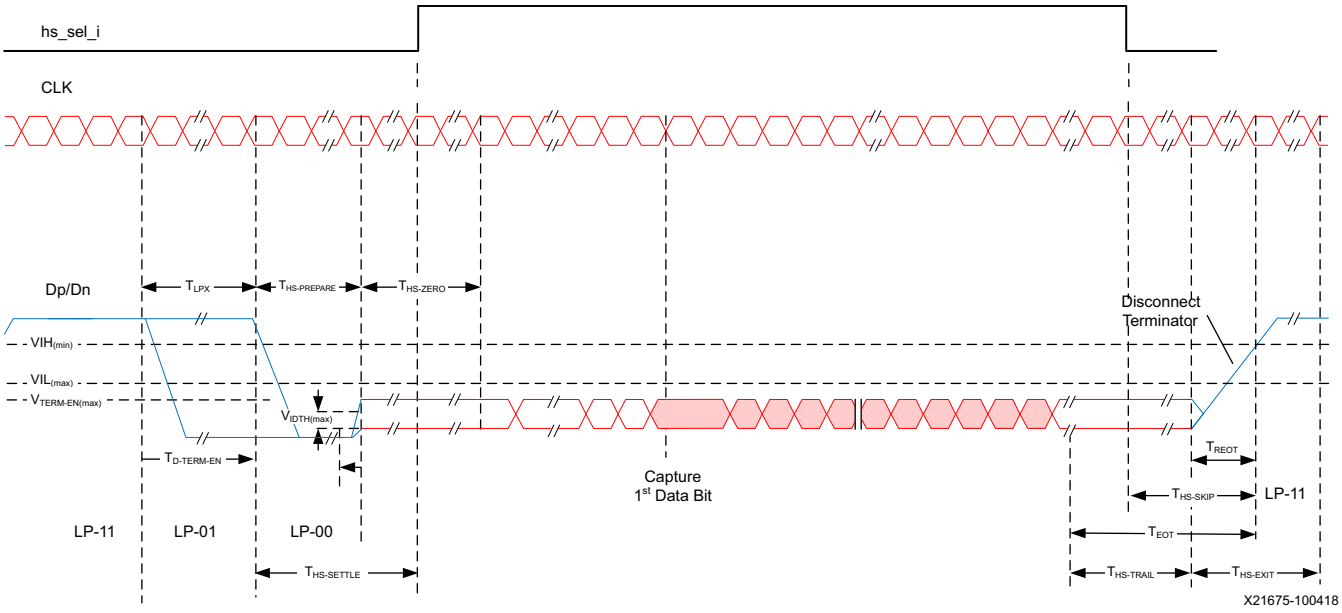


Figure 16: **HS and hs_sel_i Signal Timing at GTH Transceiver**

BIT LOCK

The PHY core logic has an interface to control the GTH transceiver, and complete a training mechanism to extract the HS datapath. After the MIPI RX line rate HS lane is stable, bit_lock goes High to indicate that the lane can be used. When all the active data lane bit_lock signals are High, the RX core logic is ready. Each MIPI data channel has one bit lock.

BIT LOSE

When the lane logic detects unstable bits from the GTH transceiver, the core logic pulls bit_lose High after a certain number of accumulation error bits within a certain period of time. This state does not go back to normal operation and needs to be reset or realigned from the controller. After reset or realign, the PHY re-locks. The bit_lose signal can work independently for each lane.

RX Clock Structure

The MIPI RX core needs a work clock with the same frequency and phase as the MIPI sending side for the FPGA to ensure that the data is received correctly. If the MIPI TX and RX sides on the PCB share a same clock source (system synchronization), the RX directly gets a clock in phase with the MIPI TX. If the clock source is different between the MIPI TX and RX, the MIPI interface TX to RX must be set to continuous clock mode. In this mode, the GTH transceiver in the HS RX clock channel outputs the recovery clock to help track the TX side clock.

Table 7: **PHY RX Pin Description**

Signal	Direction	Description
mgtrefclk	input	This is the transceiver reference clock.
gthrx/gthtx	Input/output	This is the transceiver datapath.

Table 7: **PHY RX Pin Description *(Cont'd)***

Signal	Direction	Description
rx_clk2	output	All RX side FPGA logic should work at this clock. Usually, the rx_clk2 frequency is the MIPI RX line rate divided by 16.
freerun_clk	input	This is a free running clock for the GTH transceiver DRP and other modules. It should be 50 MHz and glitch free.
rst	input	This is an asynchronous active-High reset. It resets the PHY FPGA logic and GTH transceiver.
bit_lock[3:0]	output	This is a PHY bit lock signal per lane. It means that all rxdata align well and can be used for the PCS.
bit_lose[3:0]	output	When the PHY meets an unexpected data bit and exceeds a certain number, bit_lose goes High. In this situation, the PHY requires a realign or reset to redo the bit alignment.
realign[3:0]	input	This realigns the PHY RX.
rxdata_out[63:0]	output	This is the RX data from the PHY to the MIPI PCS. There is no channel bonding between the four channels, so the PCS looks for sync sequences in each HS channel. Each MIPI HS RX lane maps to a 16-bit data bus. For example, bit[15:0] maps to HS data lane0, and bit[63:48] maps to HS data lane3.
lp_data[7:0]	output	This is the LP data from the PHY to the MIPI PCS. It is an asynchronous signal.
lp_clk[1:0]	output	This is the LP clock from the PHY to the MIPI PCS. It is an asynchronous signal.
rx_rdy	output	This is the RX ready signal from the transceiver.
hs_sel_i	input	The PCS generates this signal to help the RX PHY get the HS data.

TX

The TX PHY core logic (Figure 17) helps to combine the HS data and clock and LP data and clock to the GTH transceiver. Because a bind transceiver (TX/RX) channel code is always generated when the GTH IP is generated, `GTH_PHY_TOP.v` helps to separate the GTH RX and TX ports with the `Phy_gt_example_wrapper.v` file, and facilitates independent design of the RX 5lane and TX 5lane modules.

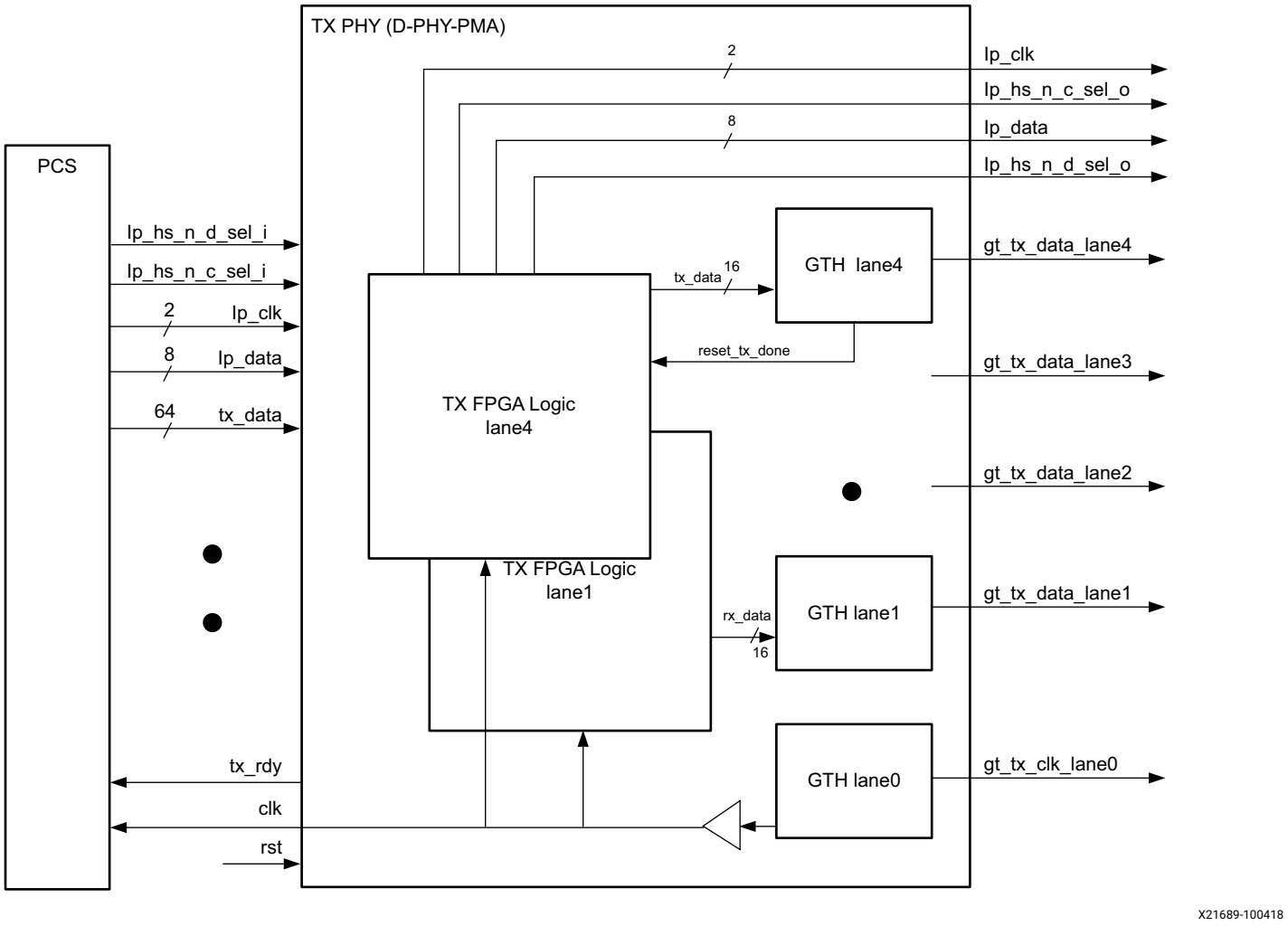


Figure 17: **2.5G MIPI D-PHY TX Logic Block**

LP Data Channel

The lp_hs_n_d_sel_i generates the lp_hs_n_d_sel_o signal for the data lane (1 for LP and 0 for HS). The signal tx_lp_data_i generates the tx_lp_data_o signal for the data lane. All input data bus signals should align well with the lp_hs_n_d_sel_i signal. The TX core logic aligns well with LP bus/HS bus/SEL relationship and then goes to the external switches.

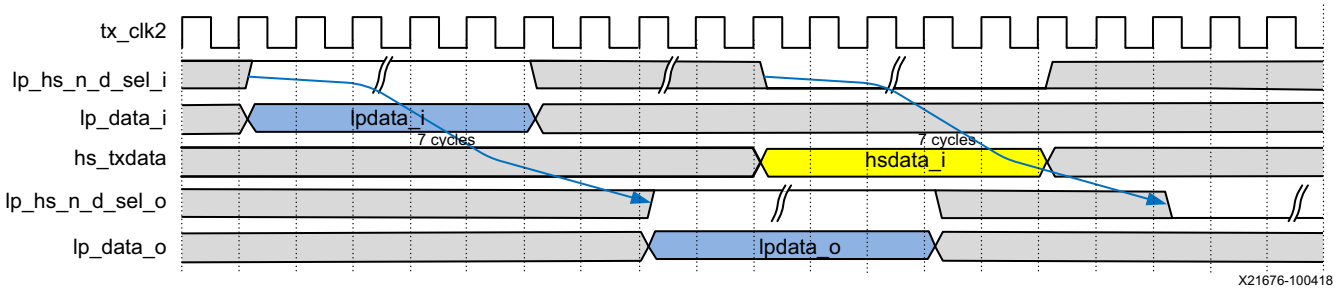


Figure 18: **2.5G MIPI D-PHY TX Interface Timing of Data Lane**

LP Clock Channel

This is the same as the datapath but with an independent port because of different switches on the board.

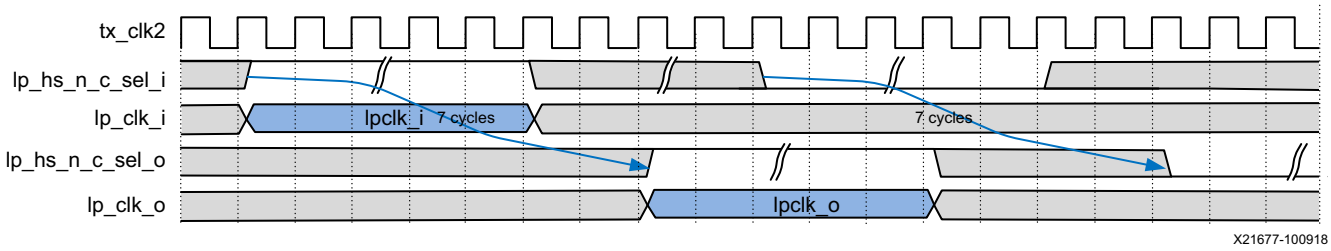


Figure 19: **2.5G MIPI D-PHY TX Interface Timing of Clock Lane**

TX Oversampling Enable

The tx_os_en signal enables TX core oversampling dependent on a variable input reference clock and MIPI lane rate. It is controlled by the multi-rate module.

HS Clock Lane Content

The tx_hs_clk_in signal helps to generate the TX HS clock signal. It receives input from the MIPI controller and has the following settings:

- `32'h0` when HS0
- `32'hFFFFFFFF` when HS1
- `32'h66666666` when clocking

Table 8: **PHY TX Pin Description**

Signal	Direction	Description
mgtrefclk	input	This is the transceiver reference clock.
gthrx/gthtx	Input/output	This is the transceiver datapath.
tx_clk2	output	The TX PCS logic should work at this clock. Usually, the tx_clk2 frequency is the MIPI TX line rate divided by 16.
rst	input	This is an asynchronous active-High reset that resets the PHY FPGA logic and GTH transceiver.
freerun_clk	input	This is a free-running clock. It should be 50 MHz from the board, by default.
txdata_out[63:0]	input	This is the TX data from the MIPI PCS to the PHY. Each MIPI HS TX lane maps to a 16-bit data bus. For example, bit[15:0] maps to HS data lane0, and bit[63:48] maps to HS data lane3.
lp_hs_n_d_sel_i	input	The lp_hs_n_d_sel_i pin generates the lp_hs_n_d_sel_o signal for the data lane. It goes from the PCS to the PHY: • 1: LP • 0: HS
lp_hs_n_d_sel_o	output	The lp_hs_n_d_sel_o pin drives the switcher directly for the data lanes from the IOB.

Table 8: **PHY TX Pin Description *(Cont'd)***

Signal	Direction	Description
lp_hs_n_c_sel_i	input	The lp_hs_n_c_sel_i pin generates the lp_hs_n_c_sel_o signal for the clock lane. It goes from the PCS to the PHY: • 1: LP • 0: HS
lp_hs_n_c_sel_o	output	The lp_hs_n_c_sel_o pin drives the switcher directly for the clock lanes from the IOB.
lp_data_i[7:0]	input	Because all data lanes share the same switcher on the board, all lanes should be kept consistent during HS and LP switching.
lp_clk_i[1:0]	input	This is the LP clock lane from the PCS to the PHY.
lp_data_o[7:0]	output	This is the LP data lane output from the PHY to the IOB.
lp_clk_o[1:0]	output	This is the LP clock lane output from the PHY to the IOB.
tx_rdy	output	This is TX ready signal from the transceiver.
tx_os_en	input	This enables TX core oversampling.

Resource Utilization

Table 9 shows the resource utilization in the reference design.

Table 9: **Resource Utilization**

Name	CLB LUTs	CLB Registers	CLBs	LUT as Logic	LUT as Memory
gth_phy_top_u (gth_phy_top)	2140	4160	666	2128	12
d_phy_gt_example_wrapper_u (d_phy_gt_example_wrapper)	558	1074	220	558	0
full_rate_ch_drp_0 (full_rate_ch_drp)	39	87	18	39	0
full_rate_ch_drp_1 (full_rate_ch_drp_3)	41	87	17	41	0
full_rate_ch_drp_2 (full_rate_ch_drp_4)	40	87	26	40	0
full_rate_ch_drp_3 (full_rate_ch_drp__parameterized0)	56	108	28	56	0
full_rate_ch_drp_4 (full_rate_ch_drp__parameterized1)	63	98	20	63	0
full_rate_ch_drp_5 (full_rate_ch_drp__parameterized1_5)	56	92	17	56	0
full_rate_ch_drp_6 (full_rate_ch_drp__parameterized0_6)	60	106	26	60	0
full_rate_ch_drp_7 (full_rate_ch_drp__parameterized1_7)	57	92	14	57	0
full_rate_cm_drp_0 (full_rate_cm_drp)	42	86	18	42	0
full_rate_cm_drp_1 (full_rate_cm_drp_8)	51	89	23	51	0
lp_tx_u (lp_tx)	12	12	3	0	12
rx_5lane_u (rx_5lane)	525	1348	214	525	0
tx_hs_5lane_os_u (tx_hs_5lane_os)	479	793	141	479	0

Multi-Line Speed Rate Support (From 100 Mb/s to 2.5 Gb/s)

The MIPI protocol provides two ways to support multi-resolution video streaming. One is to support different resolutions by encapsulating different video stream packet sizes at fixed line rates, where the interface line rate bandwidth must be larger than the maximum resolution data bandwidth. Another is that different line rates support different resolution data, while low resolution data can be supported with lower line rates.

The 2.5 Gb/s transceiver D-PHY solution also supports these two models. The advantage of fixed line rate is that it can simplify the design, and the parameters can be set into the GTH transceiver at one time without re-management, and without calling DRP units. But it is necessary to write the DRP parameter at the fixed rate to the XDC in the code.

The advantage of multi-line rate is that low resolution can be supported by a low line rate, and save more power. In multi-line rate mode, DRP units are necessary to the GTH transceiver. Users can configure the DRP through the host to adjust the transceiver parameters and adapt various line rates in real time.

Note: The reference design provides multi-line rate support by default. For fixed-line rate mode, the DRP control module can be deleted. However, the GTH attribute parameters should be written directly to the XDC, the ports should be connected to replace the port parameters in code, and the code should also be changed.

TX Line Speed Rate vs. Key Setting

According to the GTH transceiver QPLL and CPLL characteristics, the supported rates are divided into eight segments, as shown in Table 11. Set the tx_rate_sel according to the speed and choose the right PLL at the same time. Send the tx_rate_sel and txpllclksel_in results to GTH_PHY_TOP.

Table 10: **PLL Type Select**

PLL Type	txpllclksel_in	rxpllclksel_in
QPLL0	2'b11	2'b11
QPLL1	2'b10	2'b10
CPLL	2'b00	2'b00

Table 11: **TX Supported Line Speed Rates**

PLL Type	PLL Frequency Range (–2)		tx_rate_sel	Min Line Rate (Mb/s)	Max Line Rate (Mb/s)	Oversample Rate	Min PLL Frequency (MHz)	Max PLL Frequency (MHz)	TXOUT_DIV
	Min (MHz)	Max (MHz)							
QPLL1	8000	13000	7	2000	3250.0	1	8000	13000	4
			6	1000	1625.0	1	8000	13000	8
			5	500	812.5	1	8000	13000	16
			4	1600	2600.0	5	8000	13000	1
			3	800	1300.0	5	8000	13000	2
			2	400	650.0	5	8000	13000	4
			1	200	325.0	5	8000	13000	8
			0	100	162.5	5	8000	13000	16
QPLL0	9800	16375	7	2450	4093.8	1	9800	16375	4
			6	1225	2046.9	1	9800	16375	8
			5	612.5	1023.4	1	9800	16375	16
			4	1960	3275.0	5	9800	16375	1
			3	980	1637.5	5	9800	16375	2
			2	490	818.8	5	9800	16375	4
			1	245	409.4	5	9800	16375	8
			0	122.5	204.7	5	9800	16375	16
CPLL	2000	6250	7	2000	6250.0	1	2000	6250	2
			6	1000	3125.0	1	2000	6250	4
			5	500	1562.5	1	2000	6250	8
			4	800	2500.0	5	2000	6250	1
			3	400	1250.0	5	2000	6250	2
			2	200	625.0	5	2000	6250	4
			1	100	312.5	5	2000	6250	8

Notes:

1. The shaded cells show the difference between the 5x and 1x oversample rates.

- Case1: The MIPI line rate is 2.5 Gb/s

  The tx_rate_sel can choose between QPLL1 and QPLL0, or CPLL segment 7. Usually, the TX should choose segments 7–5 to cover the rate from 500 Mb/s–2.5 Gb/s, and segments 4–0 to cover 100 Mb/s–500 Mb/s.

- Case2: When the MIPI line rate is lower than 500 Mb/s

  The TX can choose 4–0 segments to cover 100 Mb/s–500 Mb/s, such as 100 Mb/s, and tx_rate_sel can choose QPLL1 segment 0 or CPLL segment 1.

RX Line Speed Rate vs. Key Setting

According to the GTH transceiver QPLL and CPLL characteristics, the supported rates are divided into five segments, as shown below. Set rx_rate_sel according to the speed and choose the right PLL at the same time. Send the rx_rate_sel and rxpllclksel_in results to GTH_PHY_TOP.

Compared to the TX side, the RX has no 7–5 segments. It has only 4–0 segments. These segments cover 100 Mb/s–2.5 Gb/s MIPI rate.

Table 12: **RX Supported Line Speed Rates**

PLL Type	PLL Frequency Range (–2)		rx_rate_sel	Min Line Rate (Mb/s)	Max Line Rate (Mb/s)	Oversample Rate	Min PLL Frequency (MHz)	Max PLL Frequency (MHz)	RXOUT_DIV
	Min (MHz)	Max (MHz)							
QPLL1	8000	13000							
			4	1600	2600.0	5	8000	13000	1
			3	800	1300.0	5	8000	13000	2
			2	400	650.0	5	8000	13000	4
			1	200	325.0	5	8000	13000	8
			0	100	162.5	5	8000	13000	16
QPLL0	9800	16375							
			4	1960	3275.0	5	9800	16375	1
			3	980	1637.5	5	9800	16375	2
			2	490	818.8	5	9800	16375	4
			1	245	409.4	5	9800	16375	8
			0	122.5	204.7	5	9800	16375	16
CPLL	2000	6250							
			4	800	2500.0	5	2000	6250	1
			3	400	1250.0	5	2000	6250	2
			2	200	625.0	5	2000	6250	4
			1	100	312.5	5	2000	6250	8

Notes:

1. The shaded cells show the difference between the 5x and 1x oversample rates.

- Case 1: The MIPI line rate is 2.5G b/s

  The rx_rate_sel can choose between QPLL1 and QPLL0, or CPLL segment 4.

- Case 2: The MIPI line rate is 100 Mb/s

  The rx_rate_sel can choose QPLL1 segment 0 or CPLL segment 1.

GTH Transceiver Reference Clock Selection and Calculation

Reference clock calculation should be based on the MIPI rate and corresponding PLL. The result of the PLL calculation (the multiple factorial coefficient) is sent to GTH_PHY_TOP to work with the reference clock.

TX QPLL Reference Clock Selection

Table 13: **TX QPLL Reference Clock Selection**

Line Rate (Mb/s)	Oversample Rate	TXOUT_DIV	QPLL Frequency (MHz)	Multiple Factorial	Reference Clock Frequency (MHz)
2500	1	4	10000	40	250
1500	1	8	12000	80	150
1200	1	8	9600	80	120
1080	1	8	8640	80	108
891	1	16	14256	80	178.2
800	1	16	12800	80	160
720	1	16	11520	80	144
640	1	16	10240	80	128
540	1	16	8640	80	108
480	5	4	9600	80	120
100	5	16	8000	80	100

Notes:

1. The shaded cells show the difference between the 5x and 1x oversample rates.

QPLL multiple factorial is selected based on the QPLL0_FBDIV/QPLL1_FBDIV attributes ranging from 16–160 in the "DRP Map of GTHE4_COMMON Primitive" table of *UltraScale Architecture GTH Transceivers User Guide* (UG576) [Ref 2]. The appropriate value should be selected based on the reference clock. The reference clock frequency calculation is given by Equation 1.

$$\text{Reference clock frequency} = \frac{\text{Line rate} \times \text{OS} \times \text{TXOUT\_DIV}}{\text{QPLL multiple factorial}} \qquad \textit{Equation 1}$$

RX QPLL Reference Clock Selection

Table 14: **RX QPLL Reference Clock Selection**

Line Rate (Mb/s)	Oversample Rate	RXOUT_DIV	QPLL Frequency (MHz)	Multiple Factorial	Reference Clock Frequency (MHz)
2500	5	1	12500	50	250
1500	5	2	15000	100	150
1200	5	2	12000	100	120

Table 14: **RX QPLL Reference Clock Selection *(Cont'd)***

Line Rate (Mb/s)	Oversample Rate	RXOUT_DIV	QPLL Frequency (MHz)	Multiple Factorial	Reference Clock Frequency (MHz)
1080	5	2	10800	100	108
891	5	2	8910	50	178.2
800	5	2	8000	50	160
720	5	4	14400	100	144
640	5	4	12800	100	128
540	5	4	10800	100	108
480	5	4	9600	80	120
100	5	16	8000	80	100

RX QPLL reference clock selection is similar to TX QPLL reference clock selection, which needs to avoid PLL conflicts. The reference clock frequency calculation is given by Equation 2.

$$\text{Reference clock frequency} = \frac{\text{Line rate} \times \text{OS} \times \text{RXOUT\_DIV}}{\text{QPLL multiple factorial}} \qquad \textit{Equation 2}$$

TX CPLL Reference Clock Selection

The CPLL multiple factorial contains the parameters CPLL_FBDIV and CPLL_FBDIV_45, based on the "DRP Map of GTHE4_CHANNEL Primitive" table in *UltraScale Architecture GTH Transceivers User Guide* (UG576) [Ref 2]. Considering these two parameters at the same time, we get M = CPLL_FBDIV × CPLL_FBDIV_45. It is necessary to select the appropriate value based on the reference clock.

Table 15: **TX CPLL Reference Clock Selection**

Line Rate (Mb/s)	Oversample Rate	TXOUT_DIV	CPLL Frequency (MHz)	Multiple Factorial	Reference Clock Frequency (MHz)
2500	1	2	2500	10	250
1500	1	4	3000	20	150
1200	1	4	2400	20	120
1080	1	4	2160	20	108
891	1	8	3564	20	178.2
800	1	8	3200	20	160
720	1	8	2880	20	144
640	1	8	2560	20	128
540	1	8	2160	20	108
480	5	2	2400	20	120
100	5	8	2000	20	100

Notes:

1. The shaded cells show the difference between the 5x and 1x oversample rates.

The reference clock frequency calculation is given by Equation 3.

$$\text{Reference clock frequency} = \frac{\text{Line rate} \times \text{OS} \times \text{TXOUT\_DIV}}{\text{CPLL\_FBDIV} \times \text{CPLL\_FBDIV\_45}} \quad \textit{Equation 3}$$

RX CPLL Reference Clock Selection

RX CPLL reference clock selection is similar to TX CPLL reference clock selection, which needs to avoid PLL conflicts.

Table 16: **RX CPLL Reference Clock Selection**

Line Rate (Mb/s)	Oversample Rate	RXOUT_DIV	CPLL Frequency (MHz)	Multiple Factorial	Reference Clock Frequency (MHz)
2500	5	1	6250	25	250
1500	5	1	3750	25	150
1200	5	1	3000	25	120
1080	5	1	2700	25	108
891	5	2	4455	25	178.2
800	5	2	4000	25	160
720	5	2	3600	25	144
640	5	2	3200	25	128
540	5	2	2700	25	108
480	5	2	2400	20	120
100	5	8	2000	20	100

The reference clock frequency calculation is given by Equation 4.

$$\text{Reference clock frequency} = \frac{\text{Line rate} \times \text{OS} \times \text{RXOUT\_DIV}}{\text{CPLL\_FBDIV} \times \text{CPLL\_FBDIV\_45}} \quad \textit{Equation 4}$$

Interface Timing

After multi-rate PLL configuration and GTH reference clock selection, the result is sent to GTH_PHY_TOP. For example, 2.5 Gb/s is shown in Figure 20.

```
//MiPi rate = 2.5G
  mgtrefclk         = clk250M;
  rxpllclksel_in    = {2'b00,2'b00,2'b00,2'b00,2'b11,2'b11,2'b11,2'b11};  // {CPLL, CPLL, CPLL, CPLL, QPLL0,QPLL0,QPLL0,QPLL0}
  txpllclksel_in    = {2'b11,2'b11,2'b11,2'b11,2'b00,2'b00,2'b00,2'b00};  // {QPLL0,QPLL0,QPLL0,QPLL0,CPLL, CPLL, CPLL, CPLL }
  cpll_fbdiv        = {8'd03,8'd03,8'd03,8'd03,8'd00,8'd00,8'd00,8'd00};  // {x5,   x5,   x5,   x5,   x2,   x2,   x2,   x2}
  cpll_fbdiv_45     = {1'b1, 1'b1, 1'b1, 1'b1, 1'b1, 1'b1, 1'b1, 1'b1 };  // {x5,   x5,   x5,   x5,   x5,   x5,   x5,   x5}
  qpll0_fbdiv       = {8'd38,8'd48};                                      // {x40,                 x50}
  qpll1_fbdiv       = {8'd38,8'd48};                                      //  NC
  rx_rate_sel       = {4'h4,4'h4,4'h4,4'h4,4'h4,4'h4,4'h4,4'h4};
  tx_rate_sel       = {4'h7,4'h7,4'h7,4'h7,4'h7,4'h7,4'h7,4'h7};
```

X21708-100918

Figure 20: **Line Rate Configuration from a Host**

Notes related to interface timing control:

1. The txpllclksel_in, rxpllclksel_in, cpll_fbdiv, cpll_fbdiv_45, qpll0_fbdiv, qpll1_fbdiv, and rate_sel signals cannot change when drp_fsm_rdy is Low. The rate change should wait for drp_fsm_rdy to go High.

2. The reconfig signal requires a High pulse of one cycle to trigger the DRP FSM.

Note: All of these parameters were configured by a host (CPU or logic state machine).

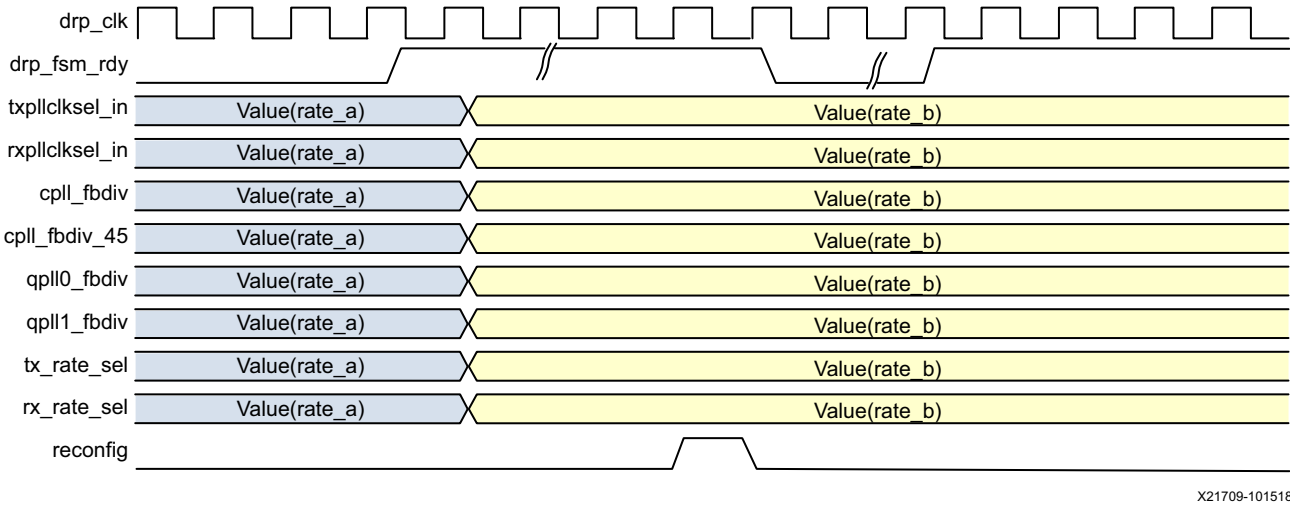


Figure 21: **Line Rate Configuration Interface Timing**

Design Requirements

Clock System Design of Receiver

Synchronous Clock System (Same Clock Source between TX and RX)

In this mode (Figure 22), the D-PHY RX does not care about the MIPI lane CLK. Thus, the MIPI CLK channel does not need to connect to the CLK lane of the RX, thereby eliminating a transceiver channel.

Note: The reference design currently does not support synchronous clock mode. An asynchronous clock mode reference design is provided to be compatible with synchronous clock mode.

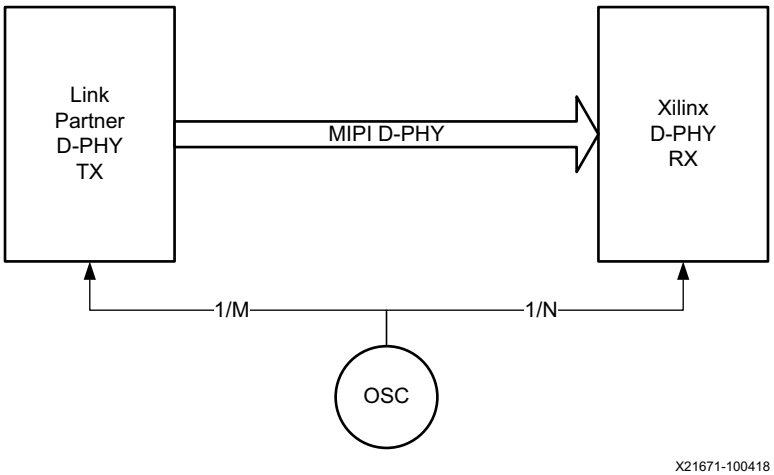


Figure 22: **Synchronous Application Mode**

For frequency selection of external reference clocks of D-PHY TX/RX, refer to Reference Clock, which requires different reference clock frequencies depending on the line rate. For phase noise, rise time, duty cycle, swing and other performance indicators, refer to the GTH Transceiver Performance table in *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS892) [Ref 6] and *Kintex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics* (DS922) [Ref 7].

Asynchronous Clock System (Independent Clock Source between TX and RX)

In this mode (Figure 23), the D-PHY RX requires that the MIPI clock work in continuous mode. Thus, the signal of the CLK channel must always work in HS mode after initialization. The CLK receiving channel of the D-PHY RX works in CDR mode. At the same time, the CDR phase parameters of the clock channel are copied to the other data channels in real time through the DRP port.(1) In this way, the DATA channel can achieve 5 times oversampling precisely, thus simplifying the design. Moreover, PCS logic is used to calibrate the HS effective time window to further ensure the stability of phase processing.

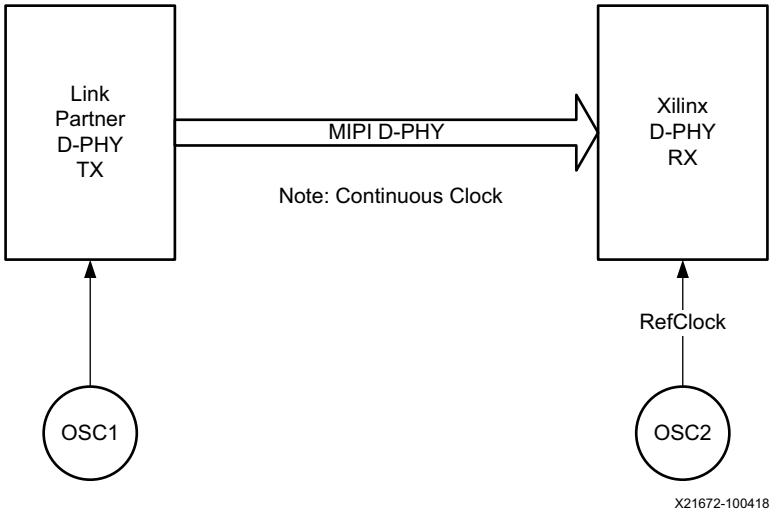


Figure 23: **Asynchronous Application Mode**

For frequency selection of external reference clocks, refer to Reference Clock, which requires different reference clock frequencies depending on the line rate. For performance indicators refer to the GTH Transceiver Switching Characteristics section in *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS892) [Ref 6] and *Kintex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics* (DS922) [Ref 7]. At the same time, the PPM deviation between OSC1 and OSC2 is not more than 200 ppm.

1. This technology is termed manual CDR-M-CDR.

I/O Assign/Timing Constraints of the 2.5G MIPI Solution

The assign/timing constraints on the TX side are as follows:

- LP (SelectIO interface):
  - The signal can be assigned to any SelectIO interface. Xilinx recommends an I/O bank of adjacent pins.
  - The LP signal must be assigned to HSUL_12_DCI for the best signal integrity (SI) and meet the MIPI standard at the same time.
  - The LP and HS SEL signal is constrained by the specific requirements of the FET switch device. (The device provided in the reference design requires a 3.3V LVCOMS level.)
  - LP_data and LP_clock follow the same regular/No I/O timing constraints for the LP signal.
- HS (GTH transceiver):
  - HS data lanes 1–4 can be assigned in a transceiver Quad. The HS_clock lane can also be assigned in the same transceiver Quad or an adjacent Quad. (The HS clock channel is transmitted through the GTH channel like data.) To support four HS data lanes, HS_clock can only be assigned to an adjacent Quad, as in the following example design.
  - Refer to the GTH design requirements in *UltraScale Architecture GTH Transceivers User Guide* (UG576) [Ref 2]. The CPLL or QPLL can be used for the HS_data lane or HS_clock lane in the GTH transceiver.

    Because the QPLL frequency coverage is limited, to support any resolution, QPLL0 and QPLL1 are occupied in a GTH Quad for the HS_data lane. The CPLL can support any resolution.
  - Refer to the example design.

On the RX side:

- LP (LP was included in the GTH transceiver):
  - There are no I/O location constraints for the LP signal.
  - Refer to the example design.
- HS (GTH transceiver):
  - The same rule applies as the MIPI TX side.

Usually, the same MIPI interface uses the same QPLL (or CPLL) for a simplified design. For designs with less than four data lanes, this is easy. Data and clock channels can be placed in one transceiver Quad. However, the reference design provides a 4+1 MIPI transmitter and receiver design to cover as many application as possible, and merges it into two transceiver Quads (Figure 24).

Four data channels occupy one Quad, and the clock channel occupies one lane in the transceiver Quad. To save clock resources, the data lanes on the TX and RX sides all use QPLL, the clock lane uses CPLL, and is thus a mixed clock design.

For the 2.5G D-PHY design transceiver clock system, you can use CPLL or QPLL, or you can mix them. This clock structure is very valuable for transceiver migration design.

Note: The transceiver TX/RX channel is generated at the same time as the GTH transceiver IP. If the TX and RX channels in the MIPI GTH both need to be used, carefully design the GTH clock and modify it based on the reference design.

The reference design is implemented on the KCU105 demo board (Kintex UltraScale KU040 FPGA) through the expansion card. The TX/RX 4 data lanes + 1 clock lane MIPI design is implemented in the two GTH Quads, respectively. The mapping of the GTH transceiver is shown in Figure 24.

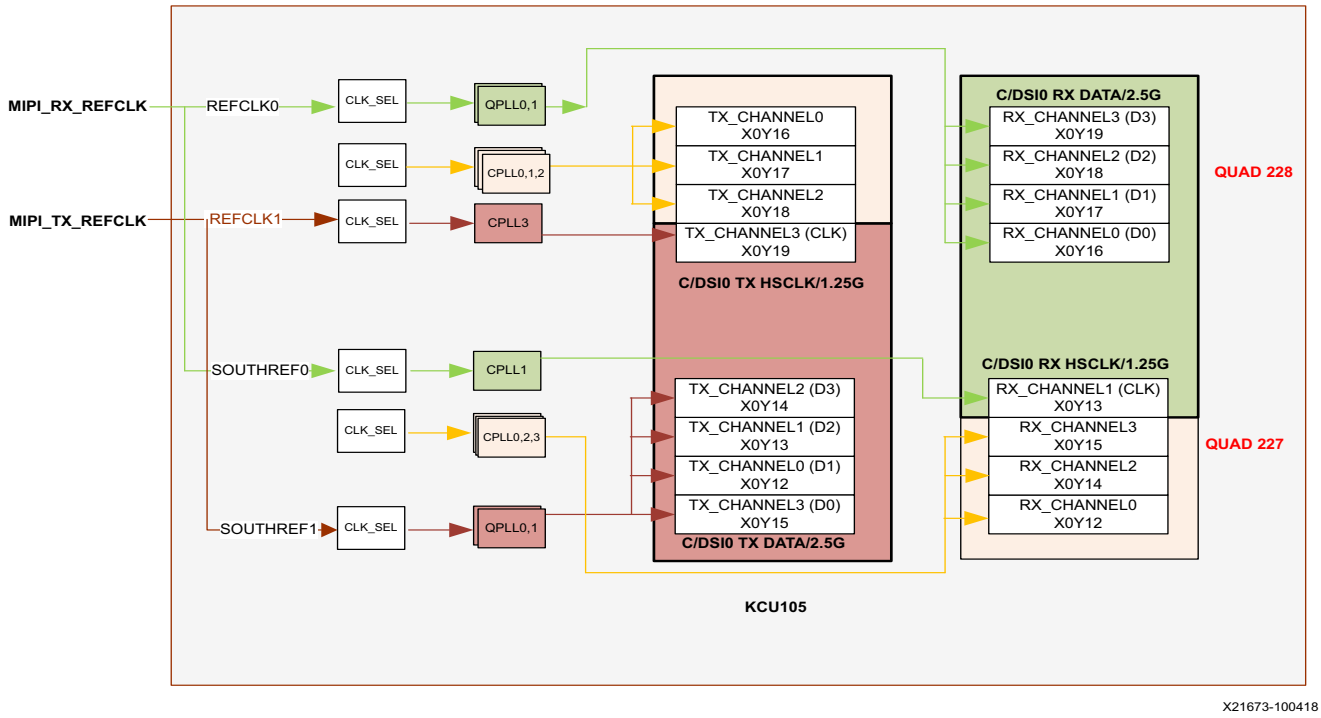


Figure 24: **Reference Design (KCU105) Transceiver Mapping + GTH Transceiver Clock Solution**

Implementing the 2.5G Transceiver D-PHY Reference Design

The GTH module is generated by a GTH Wizard. The following processes are based on the Vivado tools 2018.2.

Note: The following configuration is not directly related to physical constraints (GTH transceiver clock source and location). The user still needs to modify the code and XDC to correspond to the hardware.

The following steps describe how to regenerate the GTH IP after updating the Vivado tools version or customizing the MIPI design.

1. Open the UltraScale FPGAs transceivers Wizard in the IP catalog. In the Basic tab, set the IP (Figure 25).

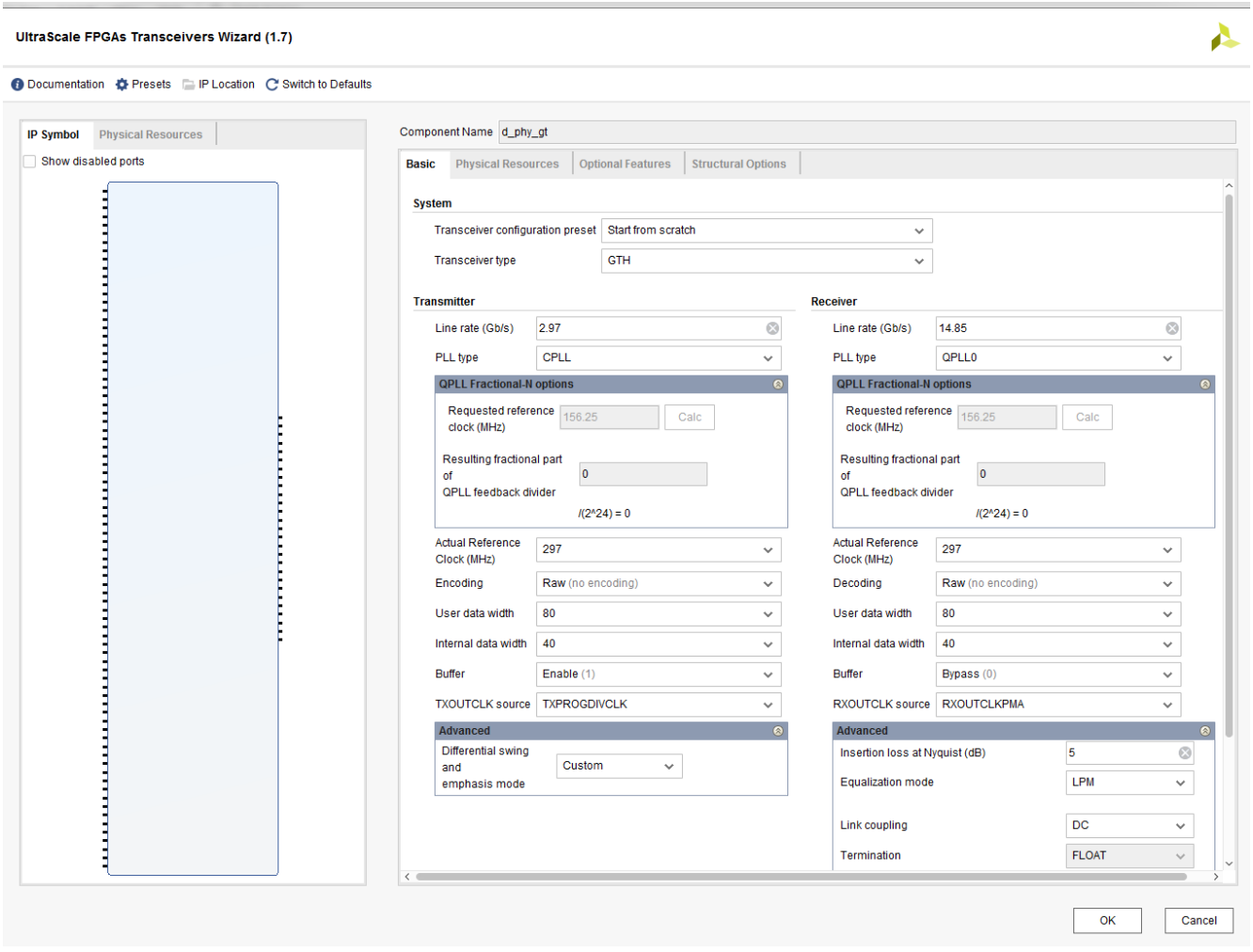


Figure 25: **Basic Tab**

2. In the Physical Resources tab, set the IP (Figure 26).

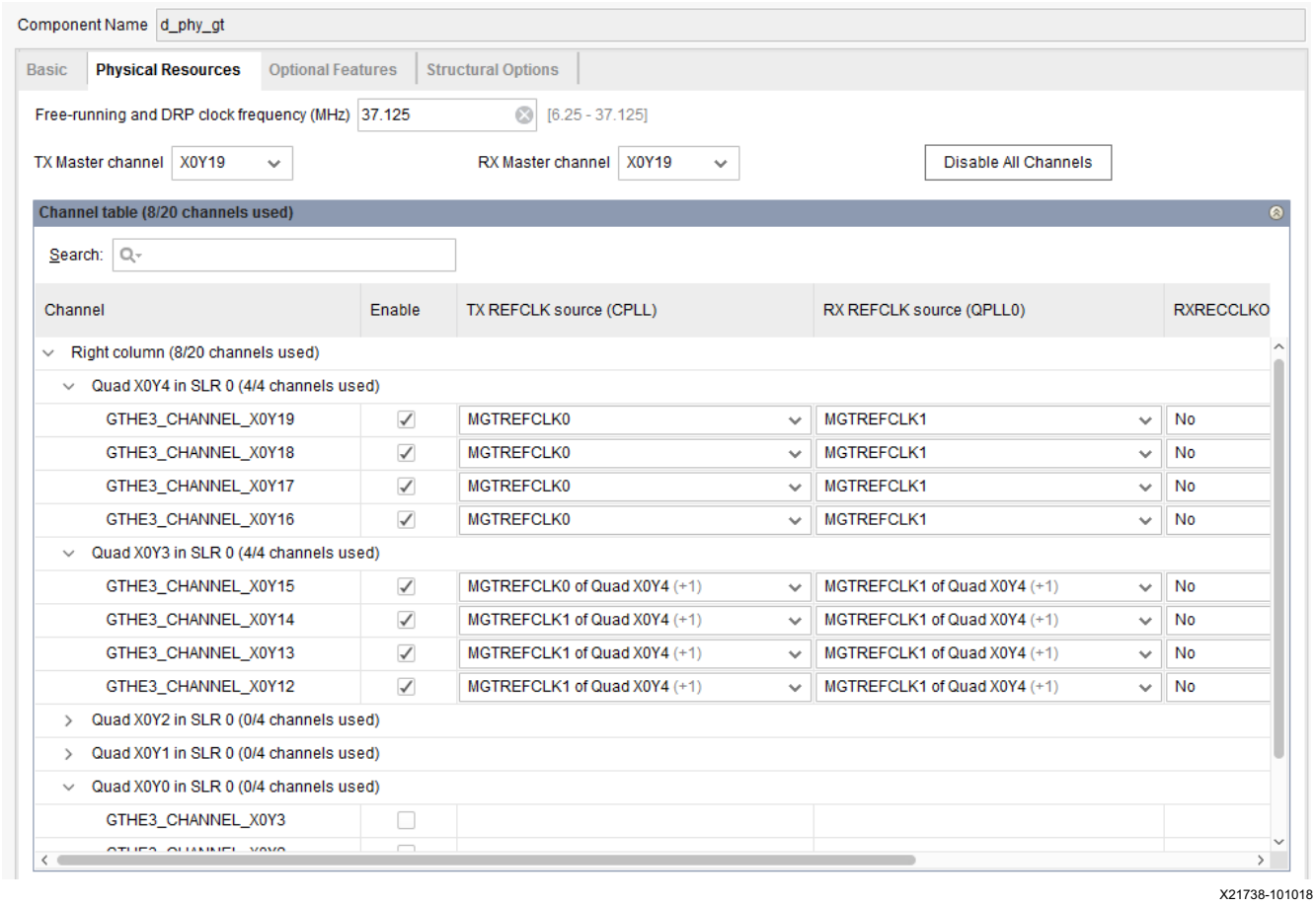


Figure 26: **Physical Resources Tab**

3. In the Optional Features tab, set the IP as follows:
   a. Set the Receiver comma detection and alignment section (Figure 27).

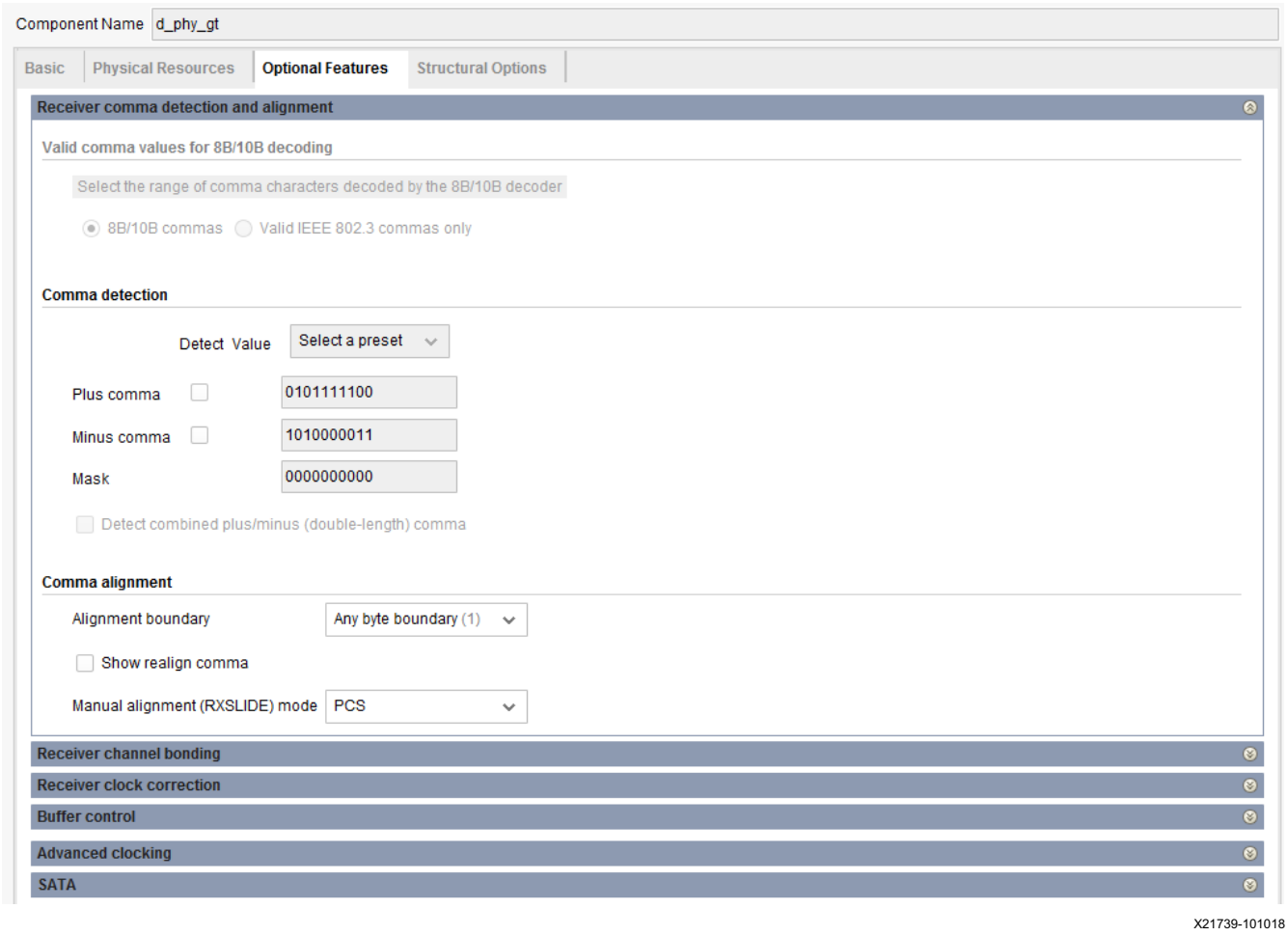


Figure 27: **Receiver Comma Detection and Alignment Section**

b. Set the Receiver channel bonding section (Figure 28).

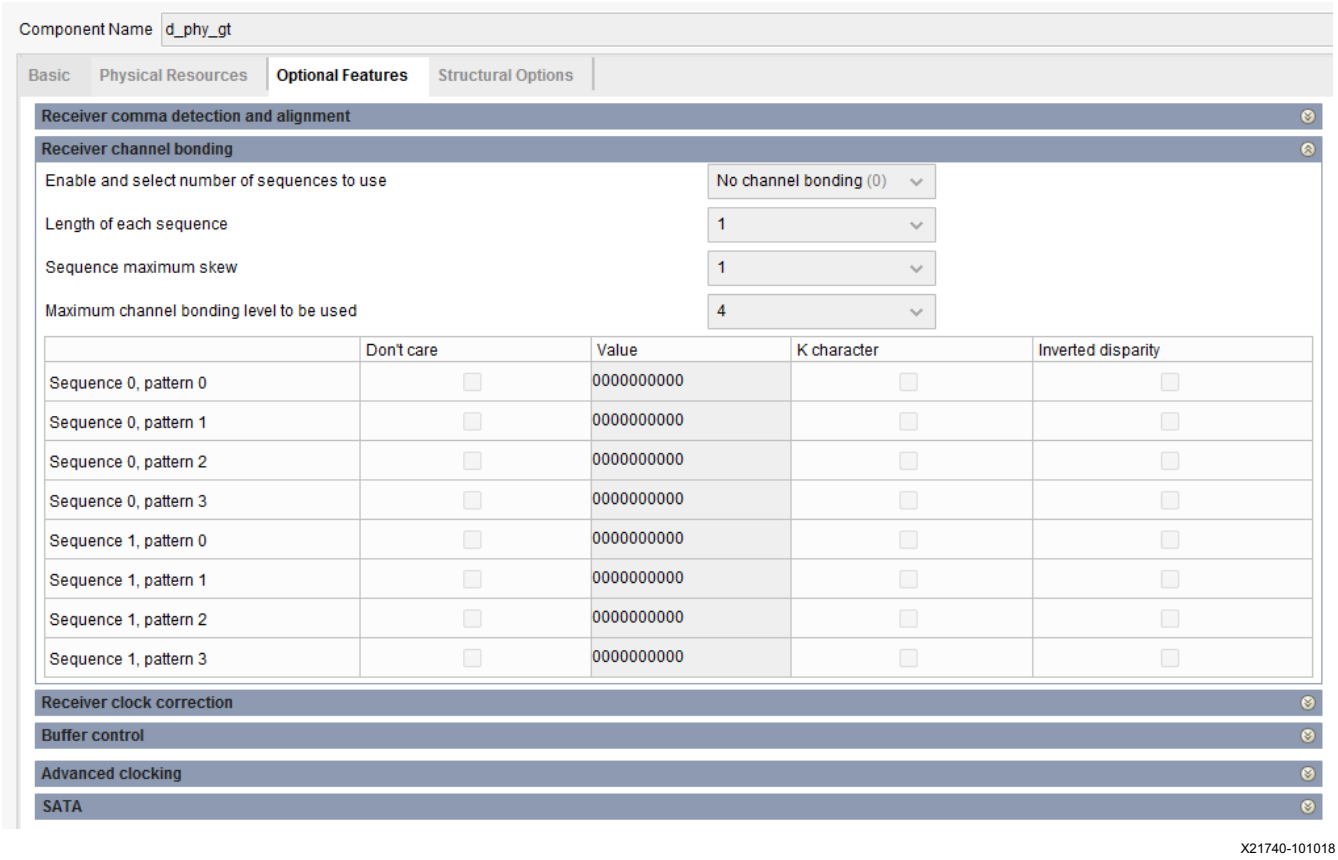


Figure 28: **Receiver Channel Bonding Section**

c. Set the Receiver clock correction section (Figure 29).

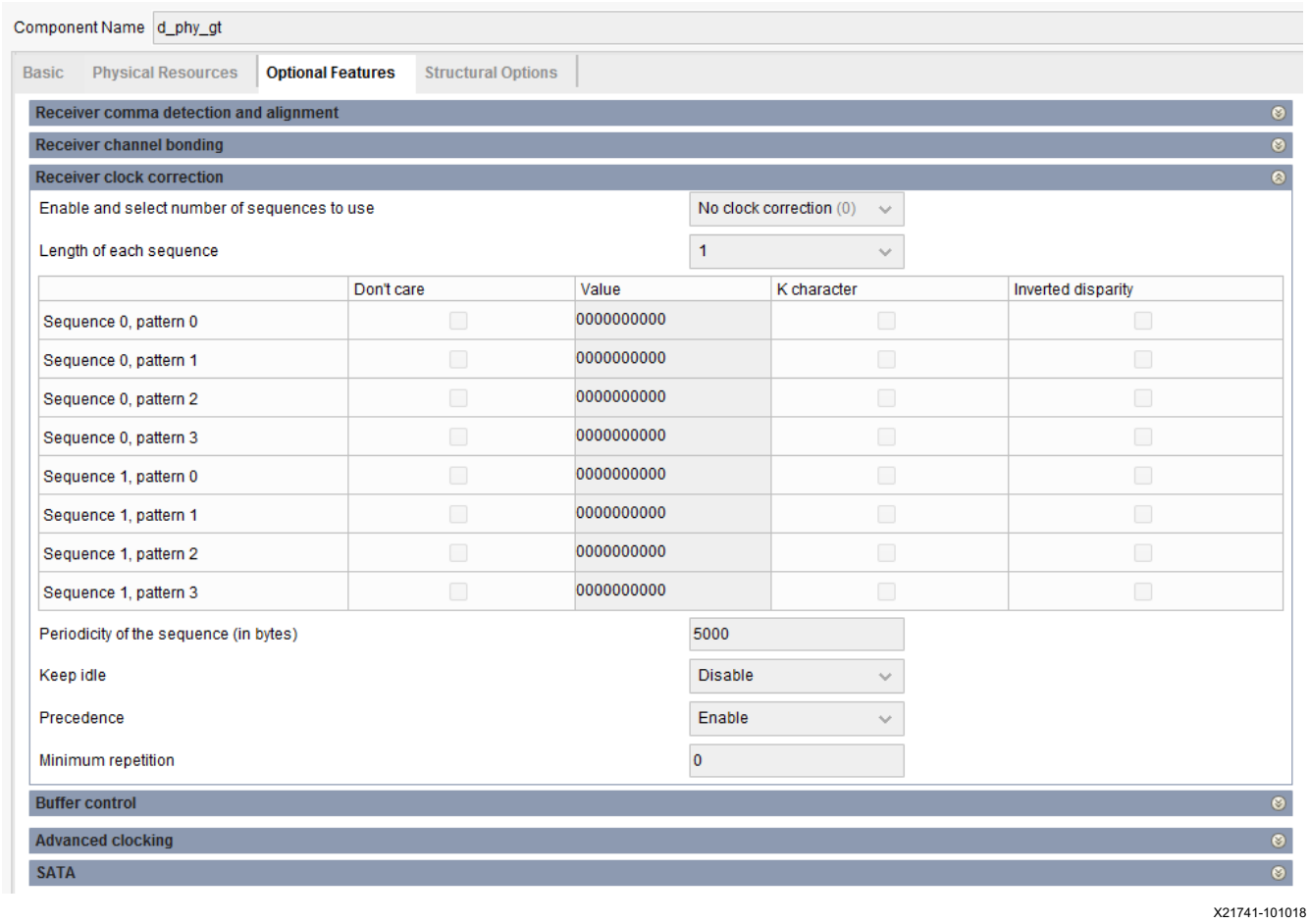


Figure 29: **Receiver Clock Correction Section**

d. Set the Buffer control section (Figure 30).

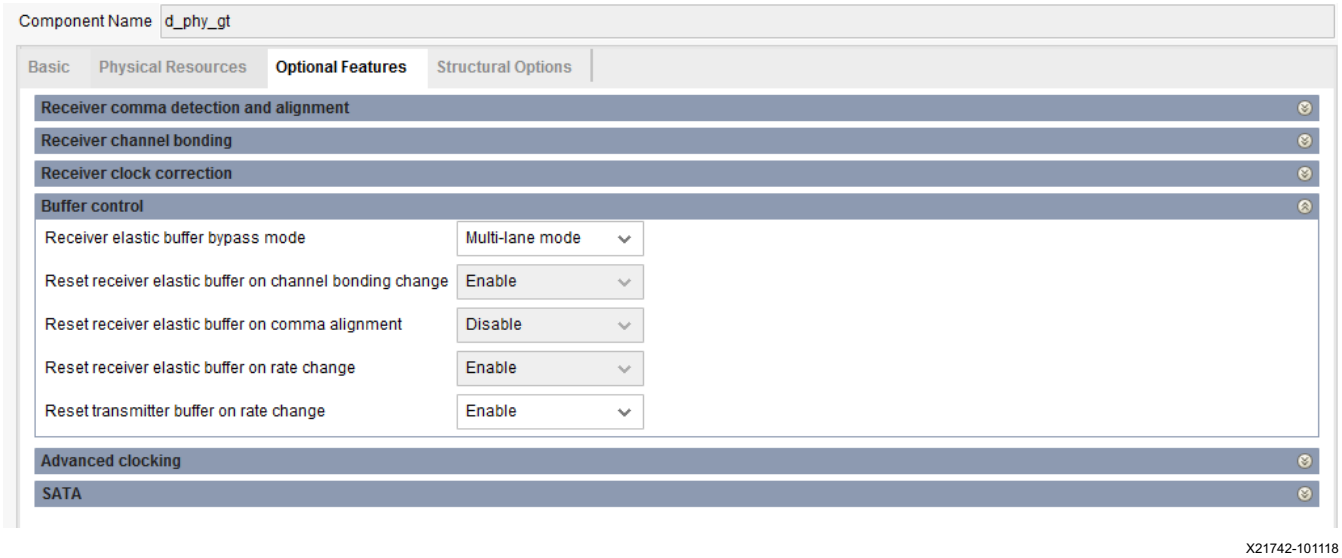


Figure 30: **Buffer Control Section**

e. Set the Advanced clocking section (Figure 31).

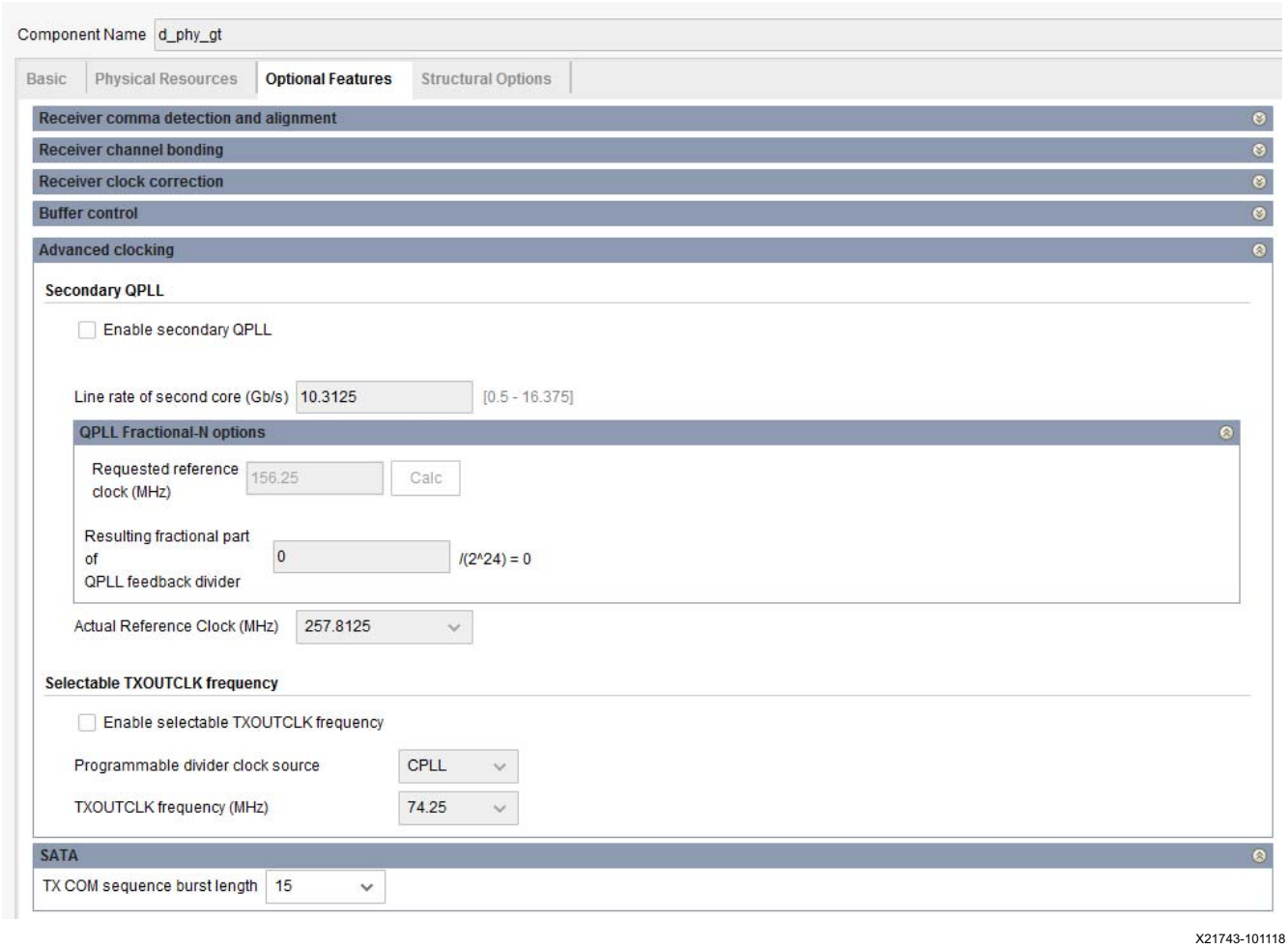


Figure 31: **Advanced Clocking Section**

4. In the Structural Options tab, set the IP as follows:
   a. Set the Structural Options section (Figure 32).

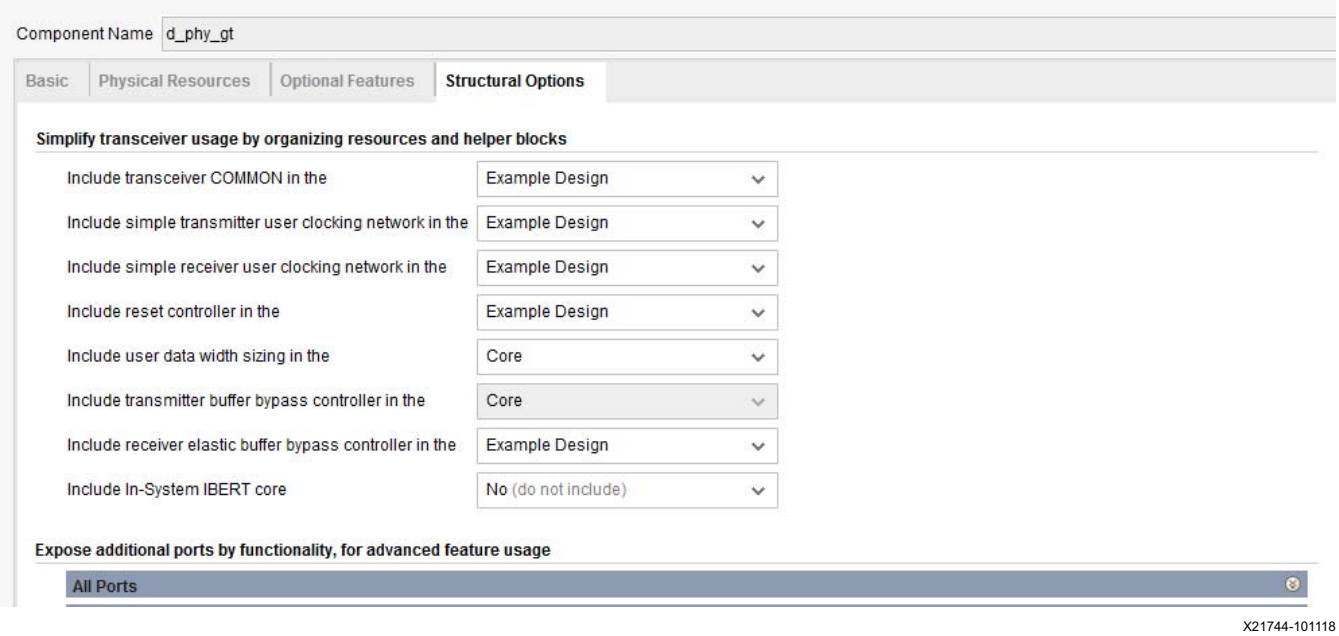


Figure 32: **Structural Options Section**

b. Set the Transceiver-based IP Debug Ports section (Figure 33).

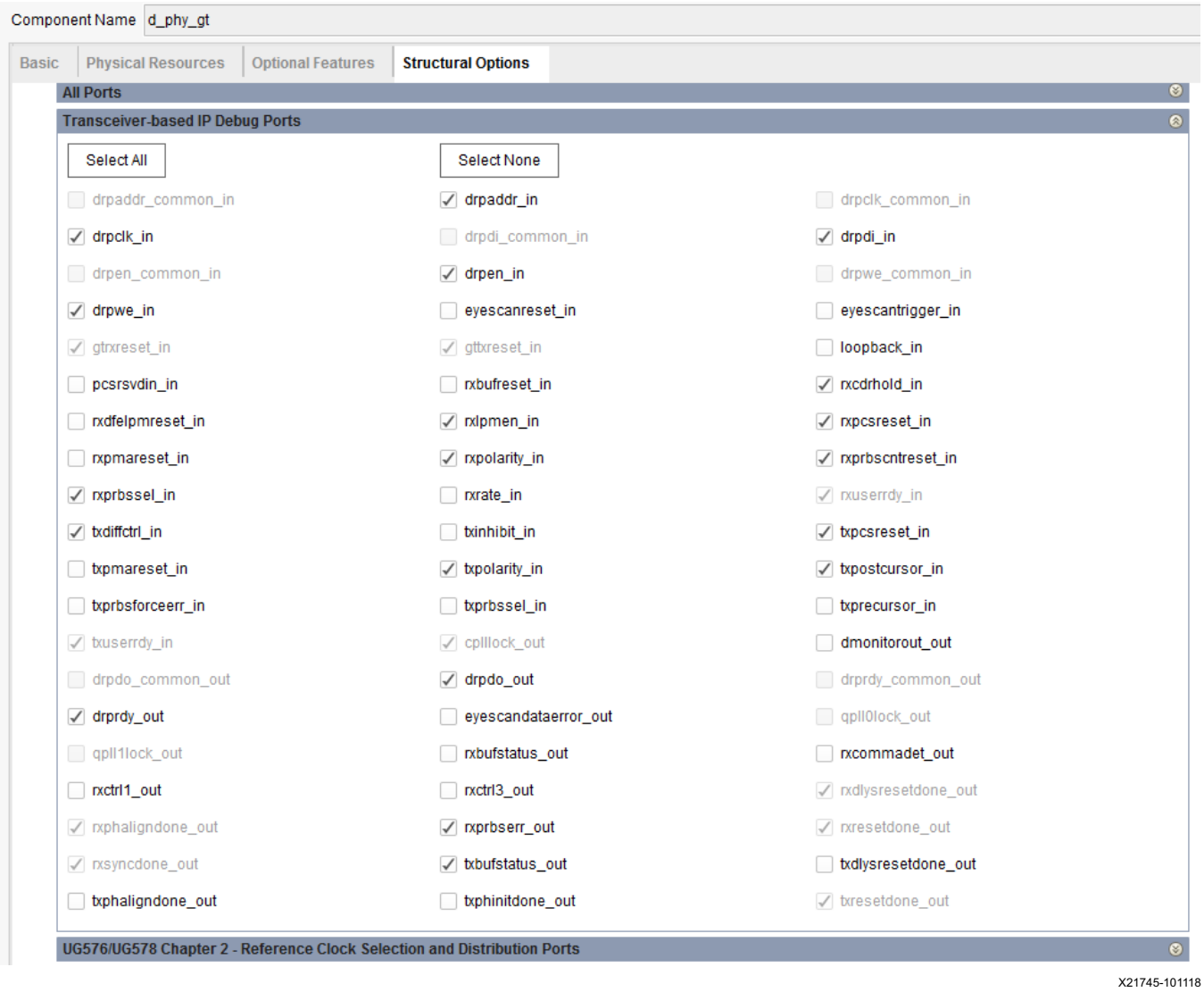


Figure 33: **Transceiver-based IP Debug Ports Section**

c. Set the UG576/UG578 Chapter 2 - Reference Clock Selection and Distribution Ports section (Figure 34).

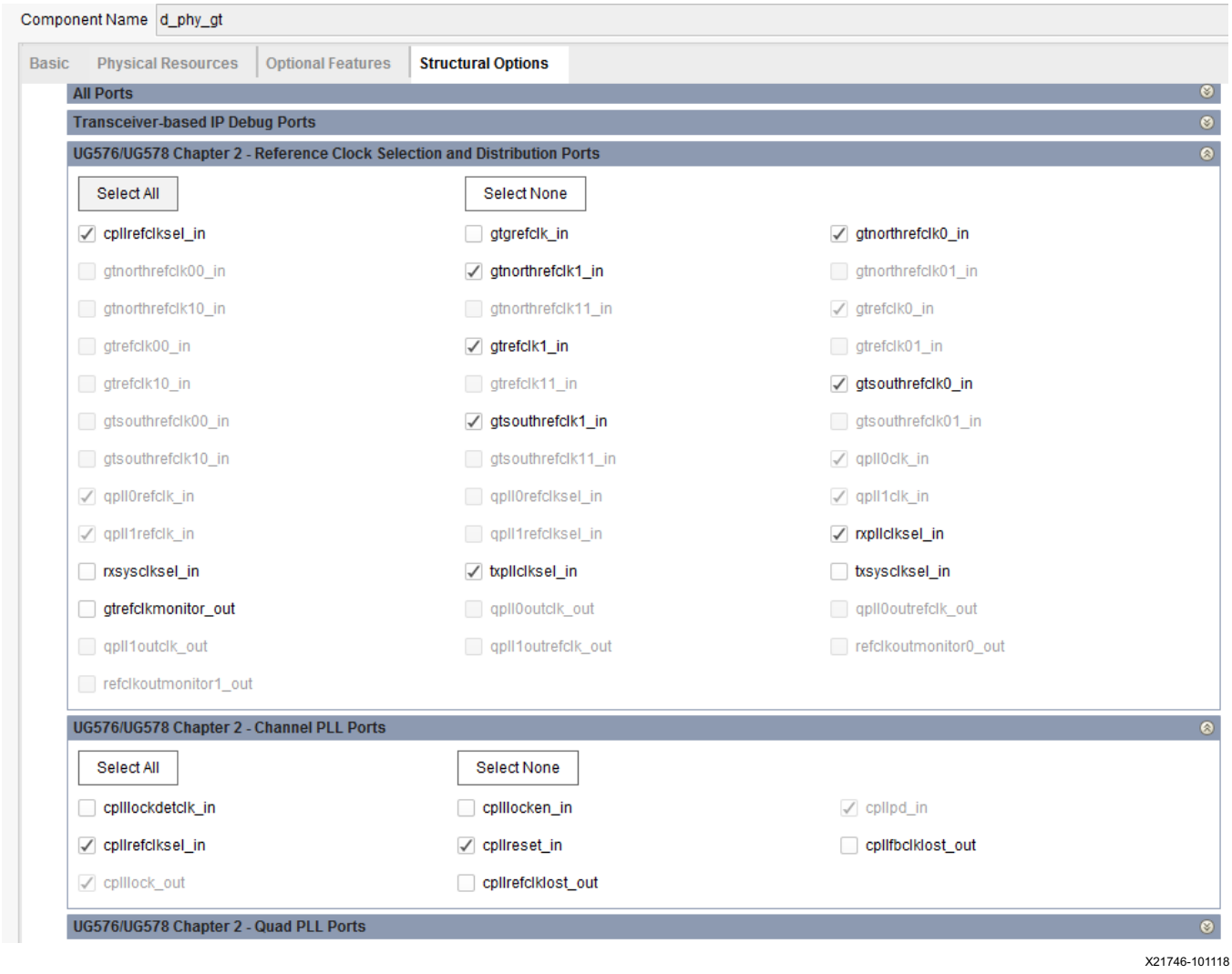


Figure 34: **UG576/UG578 Chapter 2 - Reference Clock Selection and Distribution Ports Section**

d. Set the UG576/UG578 Chapter 2 - Quad PLL Ports, Reset and Initialization Ports, and CPLL Reset Ports sections (Figure 35).

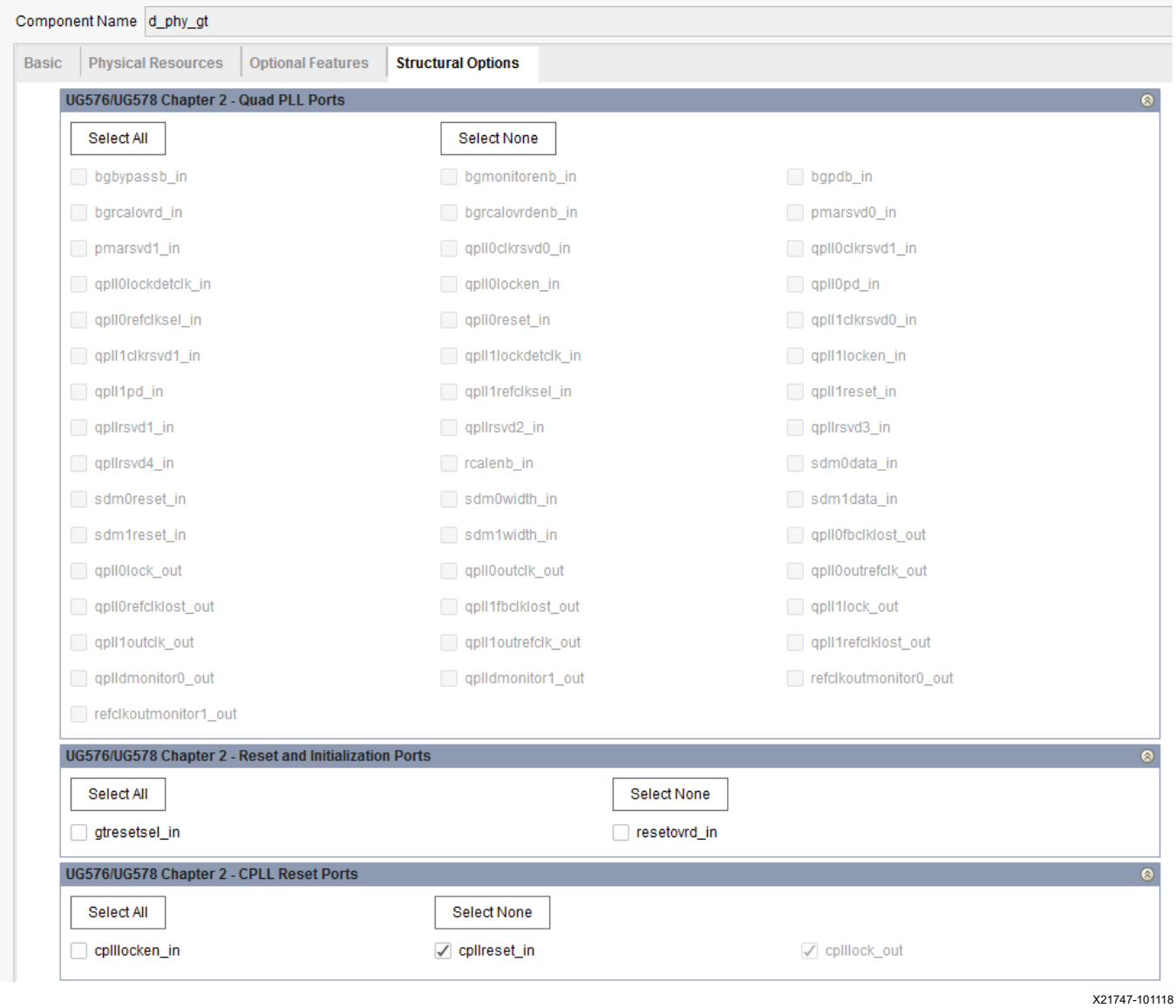


Figure 35: **UG576/UG578 Chapter 2 - Quad PLL Ports, Reset and Initialization Ports, and CPLL Reset Ports Sections**

e. Set the UG576/UG578 Chapter 2 - QPLL0/1 Reset Ports, TX Initialization and Reset Ports, RX Initialization and Reset Ports, and Power Down Ports sections (Figure 36).

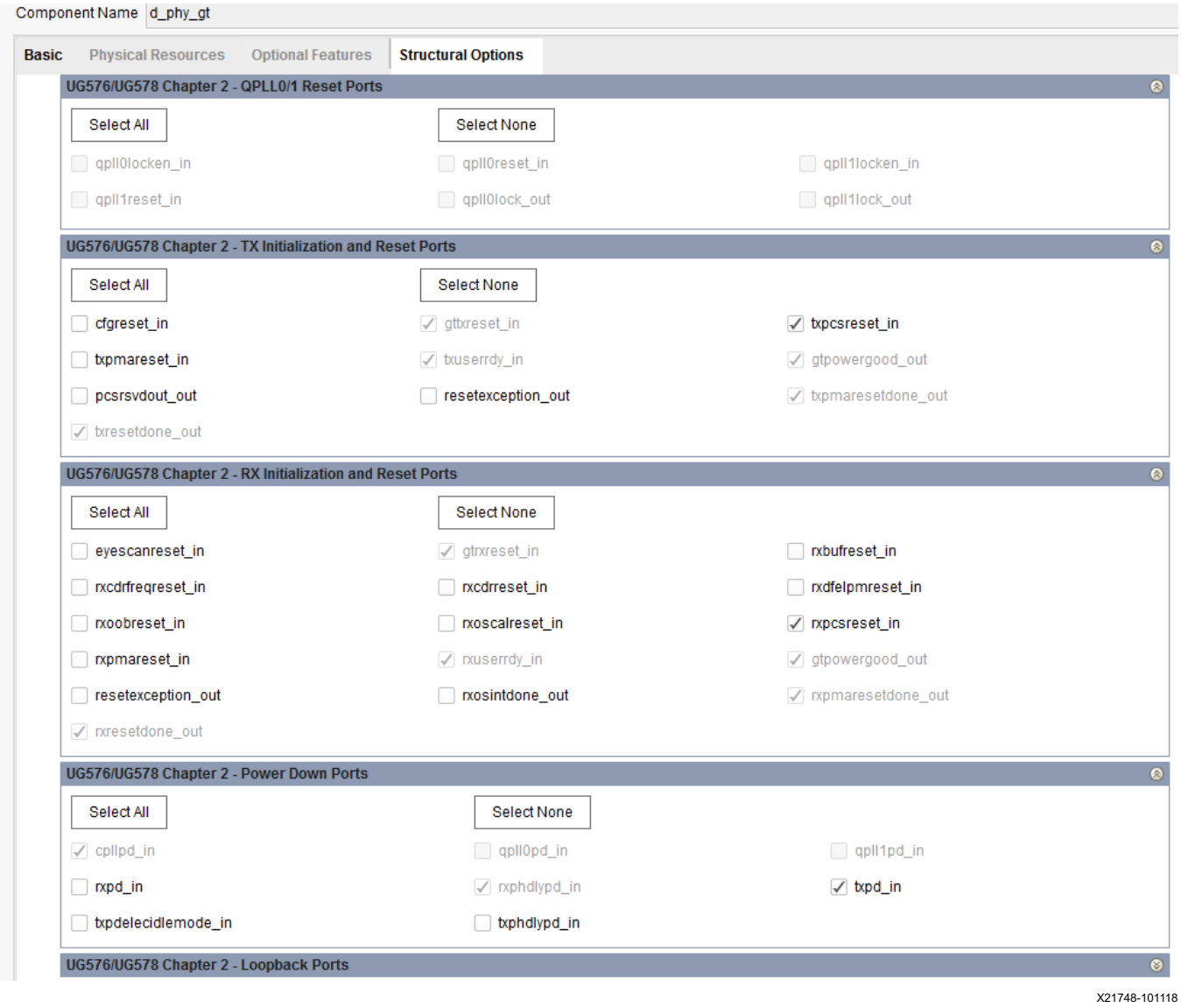


Figure 36: **UG576/UG578 Chapter 2 - QPLL0/1 Reset Ports, TX Initialization and Reset Ports, RX Initialization and Reset Ports, and Power Down Ports Sections**

f. Set the UG576/UG578 Chapter 2 - Loopback ports, Dynamic Reconfiguration Ports, FPGA TX Interface Ports, TX 8B/10B Encoder Ports, and TX Synchronous Gearbox Ports sections (Figure 37).

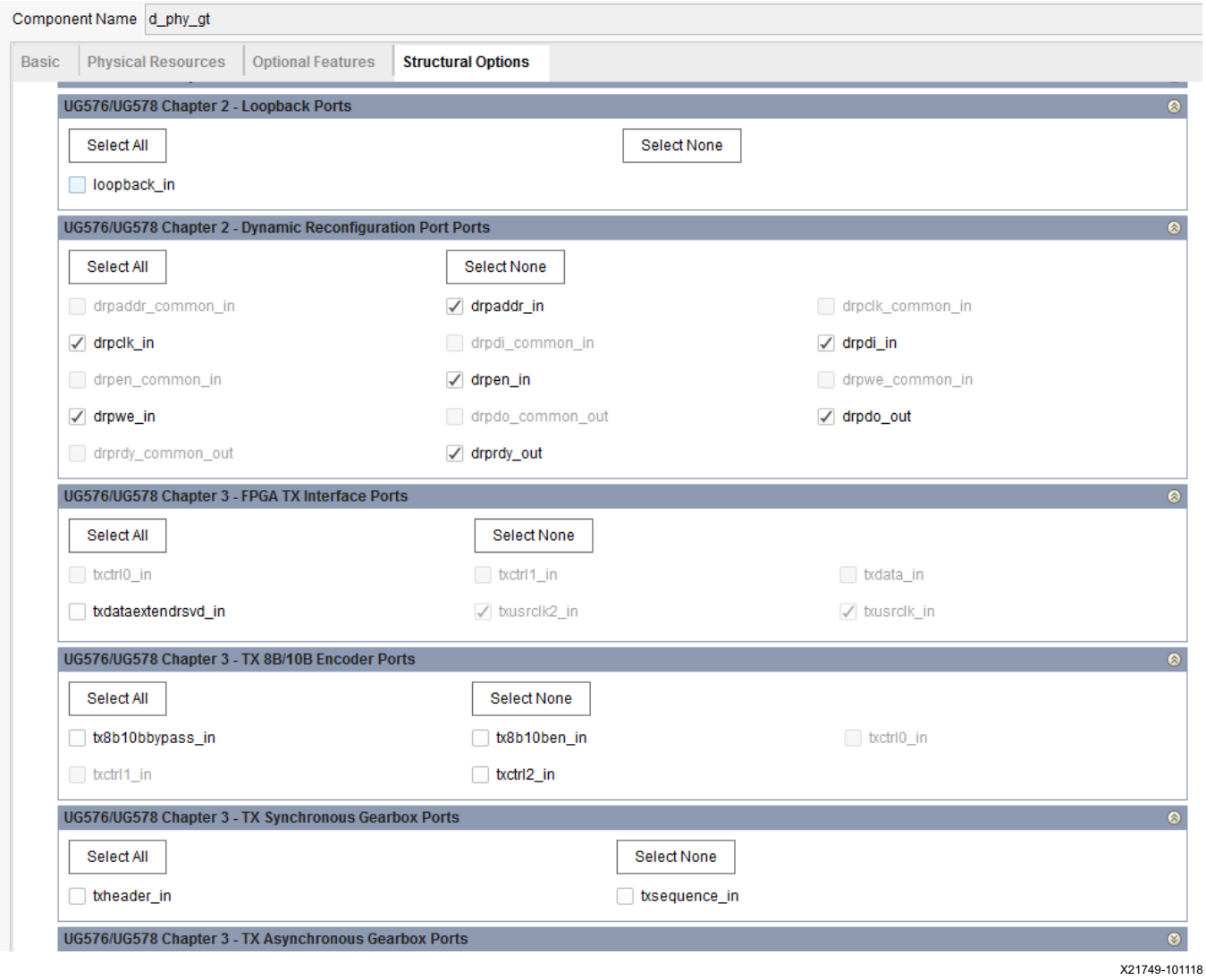


Figure 37: **UG576/UG578 Chapter 2 - Loopback ports, Dynamic Reconfiguration Ports, FPGA TX Interface Ports, TX 8B/10B Encoder Ports, and TX Synchronous Gearbox Ports Sections**

g. Set the UG576/UG578 Chapter 3 - TX Asynchronous Gearbox Ports, TX Buffer Ports, TX Pattern Generator Ports, and TX Polarity Control Ports sections (Figure 38).

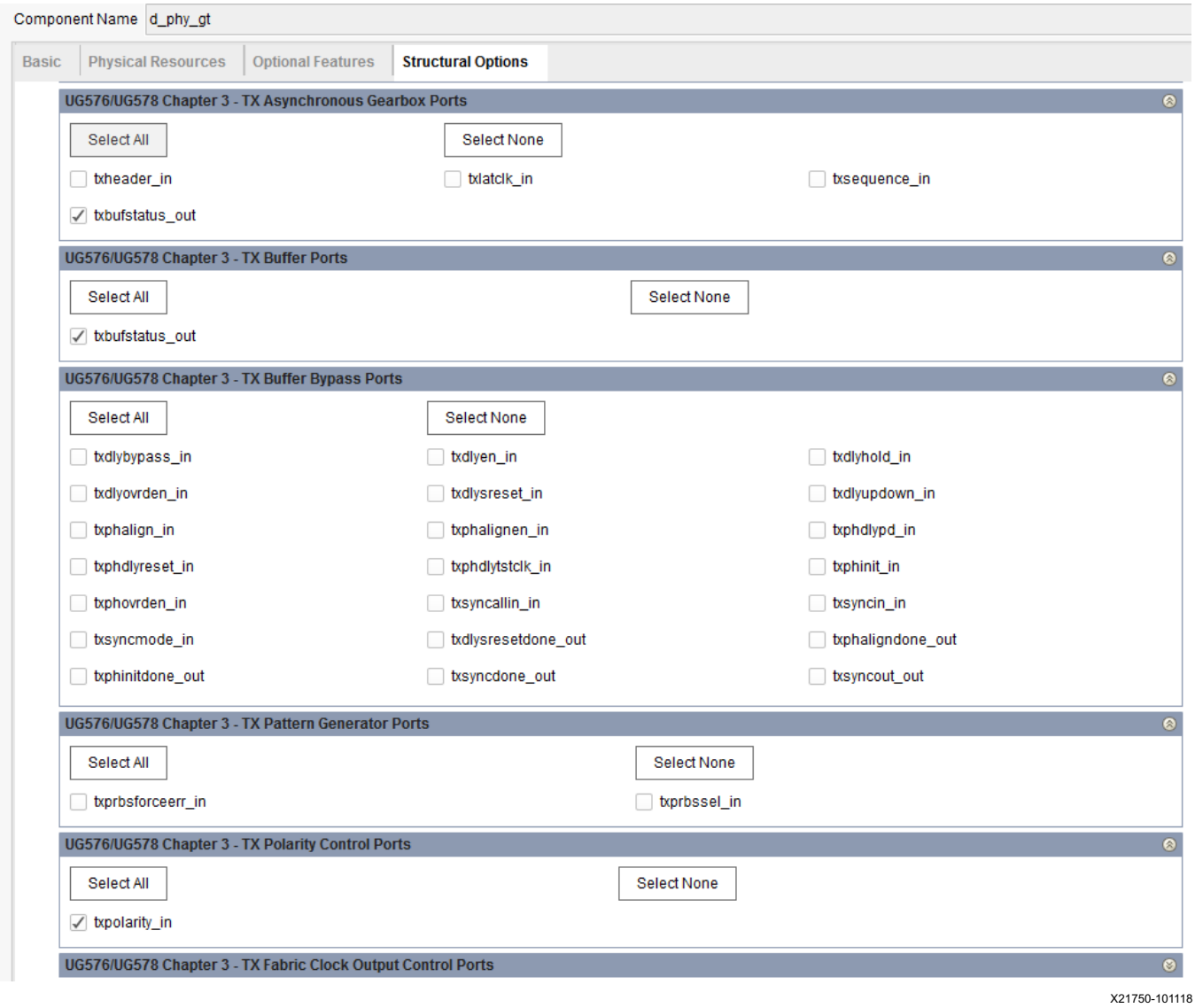


Figure 38: **UG576/UG578 Chapter 3 - TX Asynchronous Gearbox Ports, TX Buffer Ports, TX Pattern Generator Ports, and TX Polarity Control Ports Sections**

h. Set the UG576/UG578 Chapter 3 - TX Fabric Clock Output Control Ports, TX Phase Interpolator PPM Controller Ports, and TX Configurable Driver Ports sections (Figure 39).

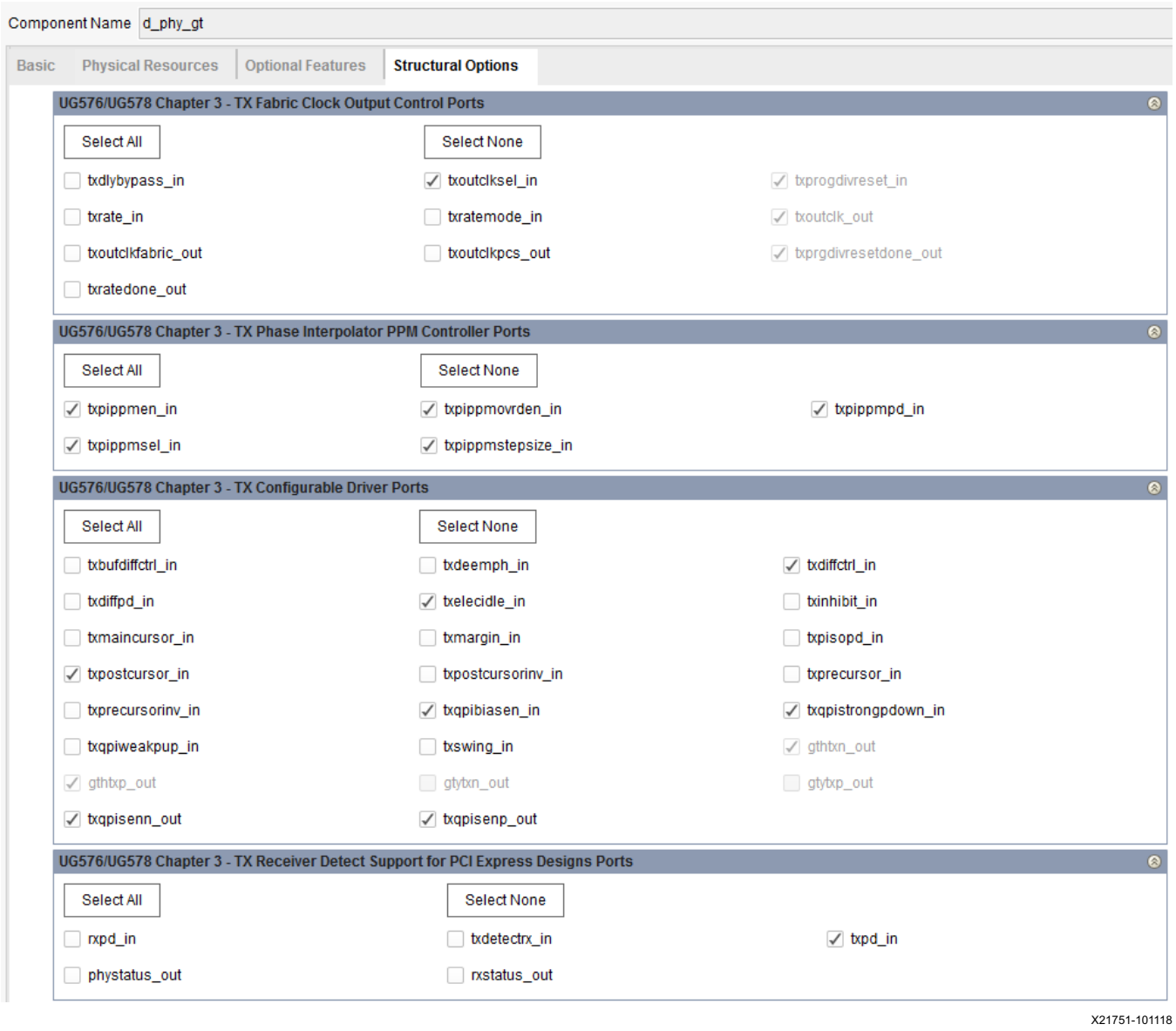


Figure 39: **UG576/UG578 Chapter 3 - TX Fabric Clock Output Control Ports, TX Phase Interpolator PPM Controller Ports, and TX Configurable Driver Ports Sections**

i. Set the UG576/UG578 Chapter 3 - TX Out-of-Band Signaling Ports and Chapter 4 - RX Analog Front End Ports sections (Figure 40).

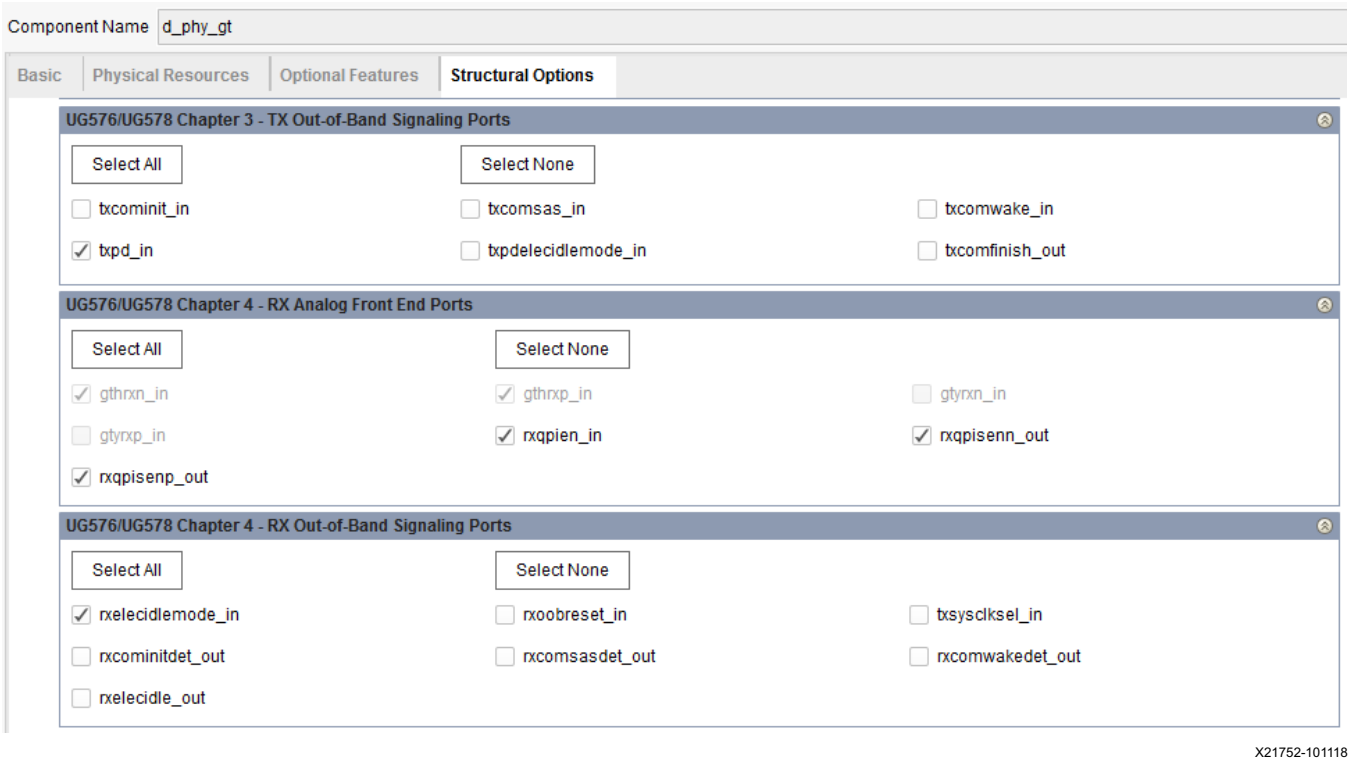


Figure 40: **UG576/UG578 Chapter 3 - TX Out-of-Band Signaling Ports and Chapter 4 - RX Analog Front End Ports Sections**

j. Set the UG576/UG578 Chapter 4 - RX Equalizer (DFE and LPM) Ports section (Figure 41).

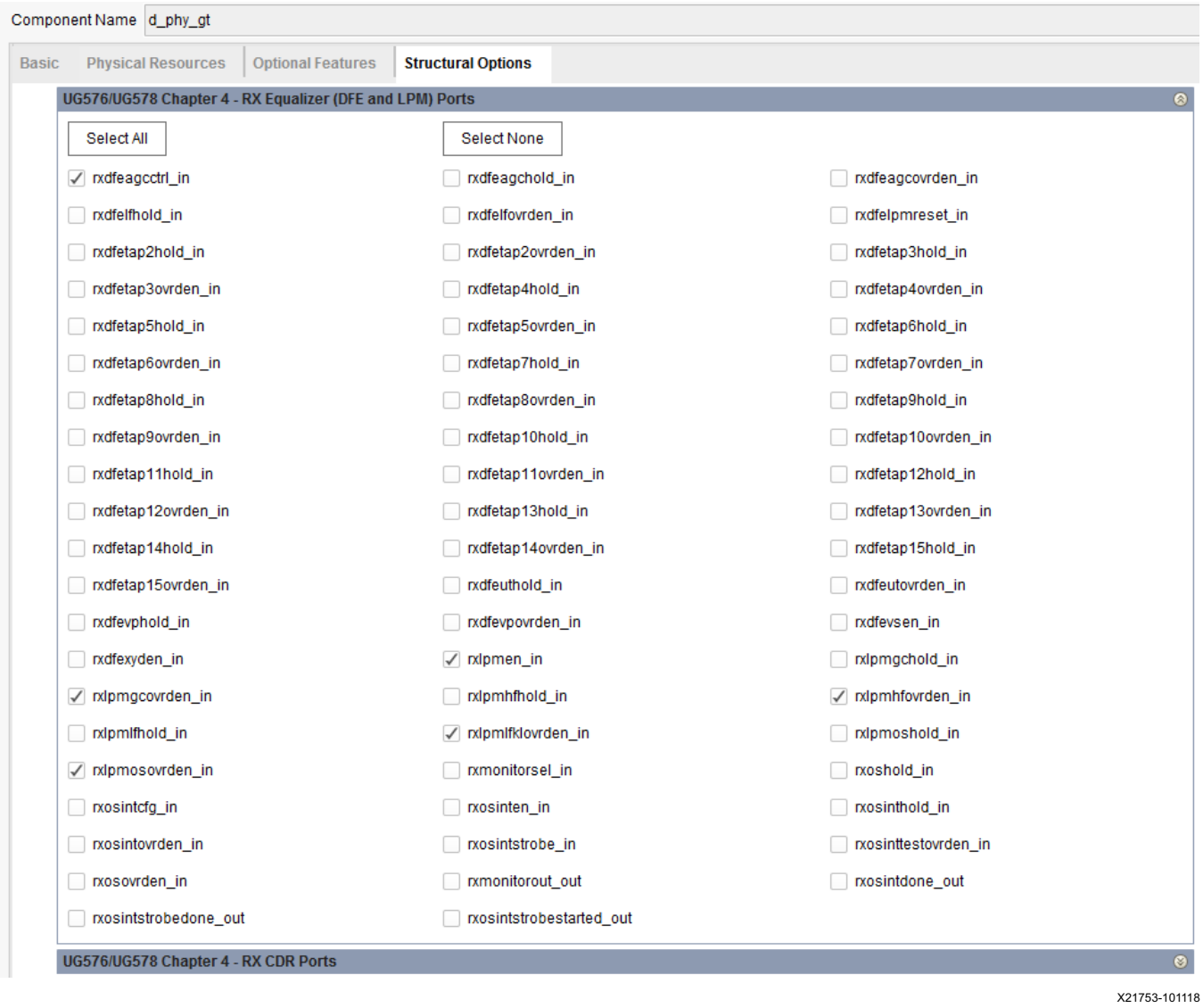


Figure 41: **UG576/UG578 Chapter 4 - RX Equalizer (DFE and LPM) Ports Section**

k. Set the Chapter 4 - RX CDR Ports section (Figure 42).

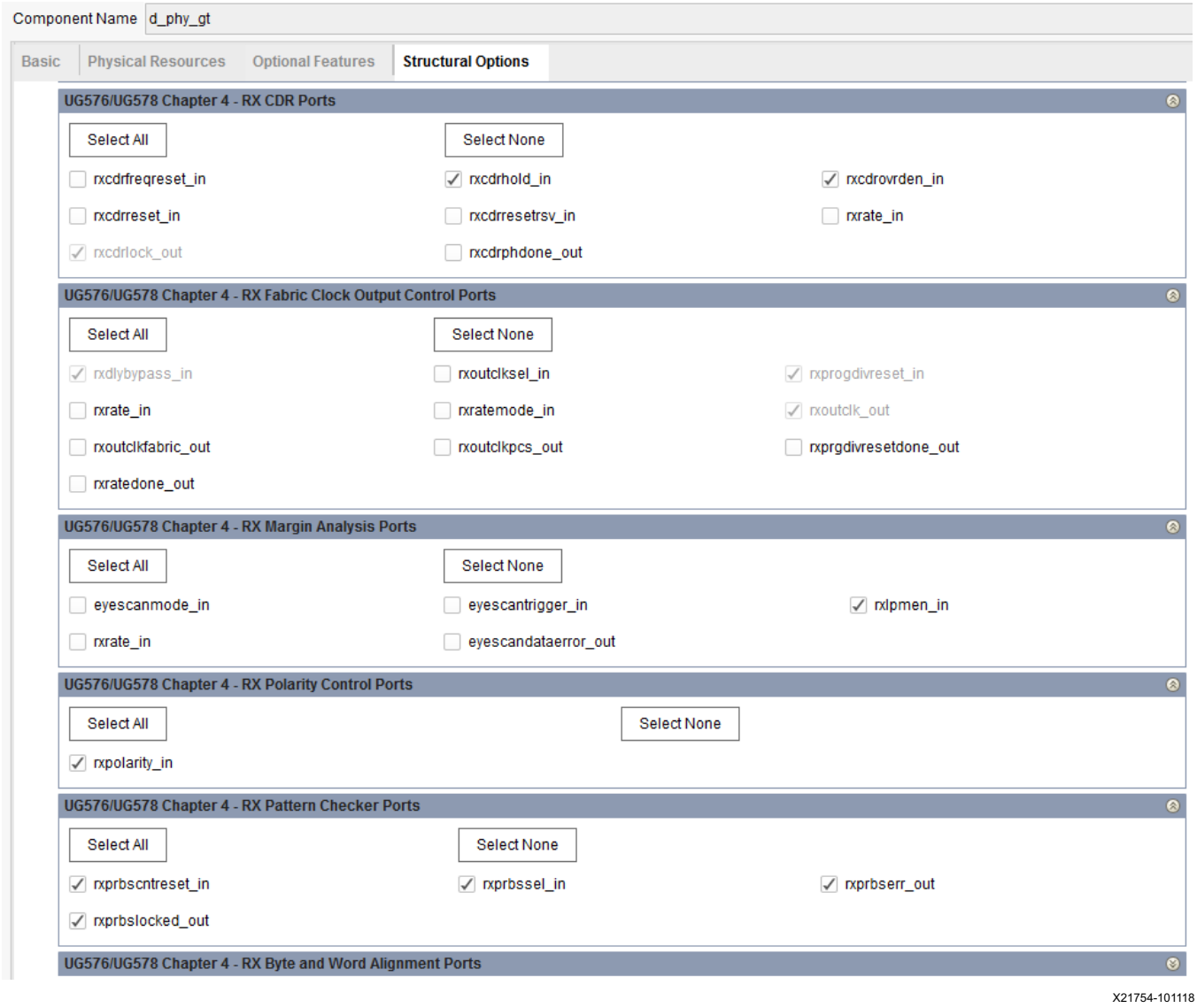


Figure 42: **Chapter 4 - RX CDR Ports Section**

l. Set the Chapter 4 - RX Byte and Word Alignment Ports section (Figure 43).

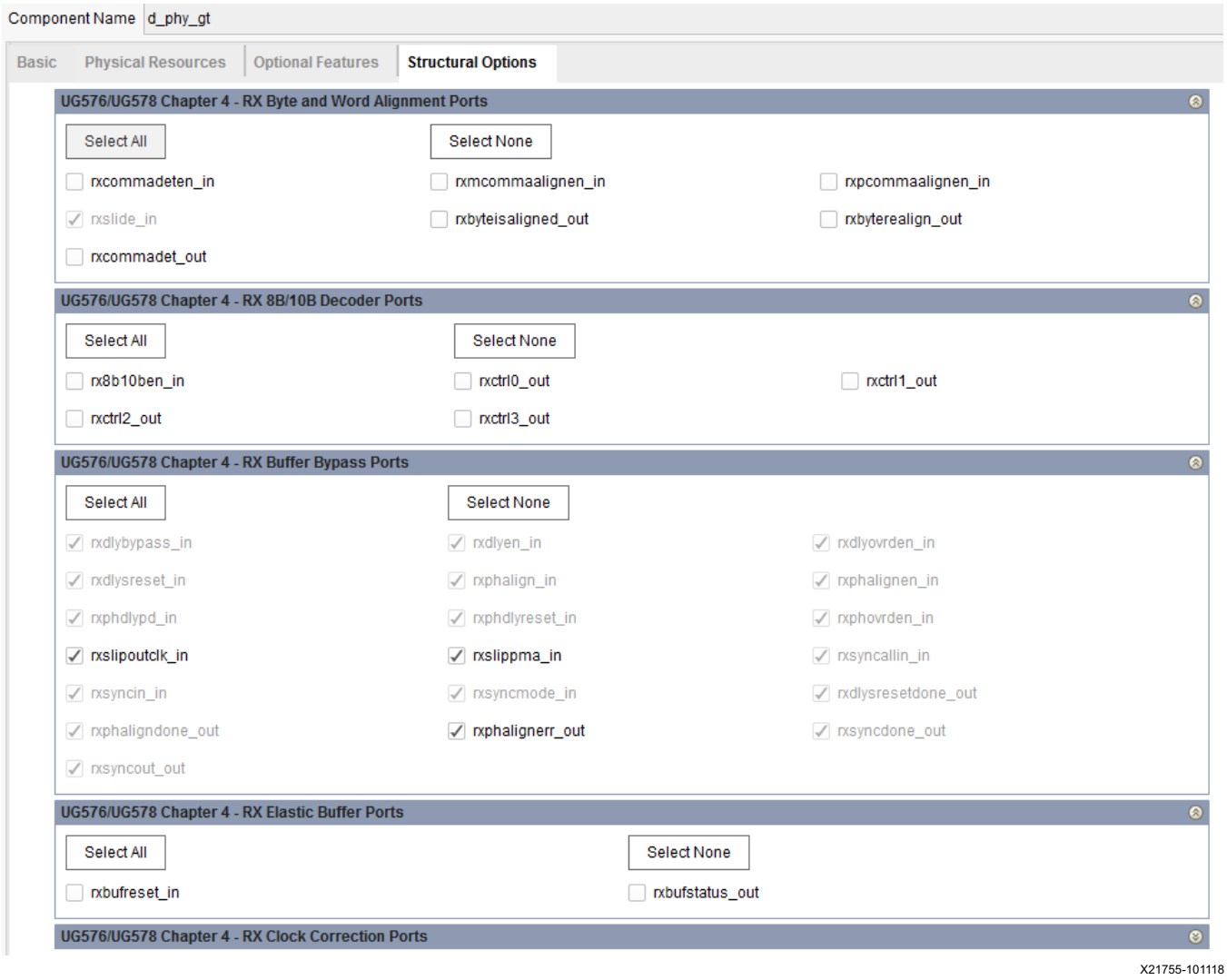


Figure 43: **Chapter 4 - RX Byte and Word Alignment Ports Section**

m. Set the Chapter 4 - RX Clock Correction Ports section (Figure 44).

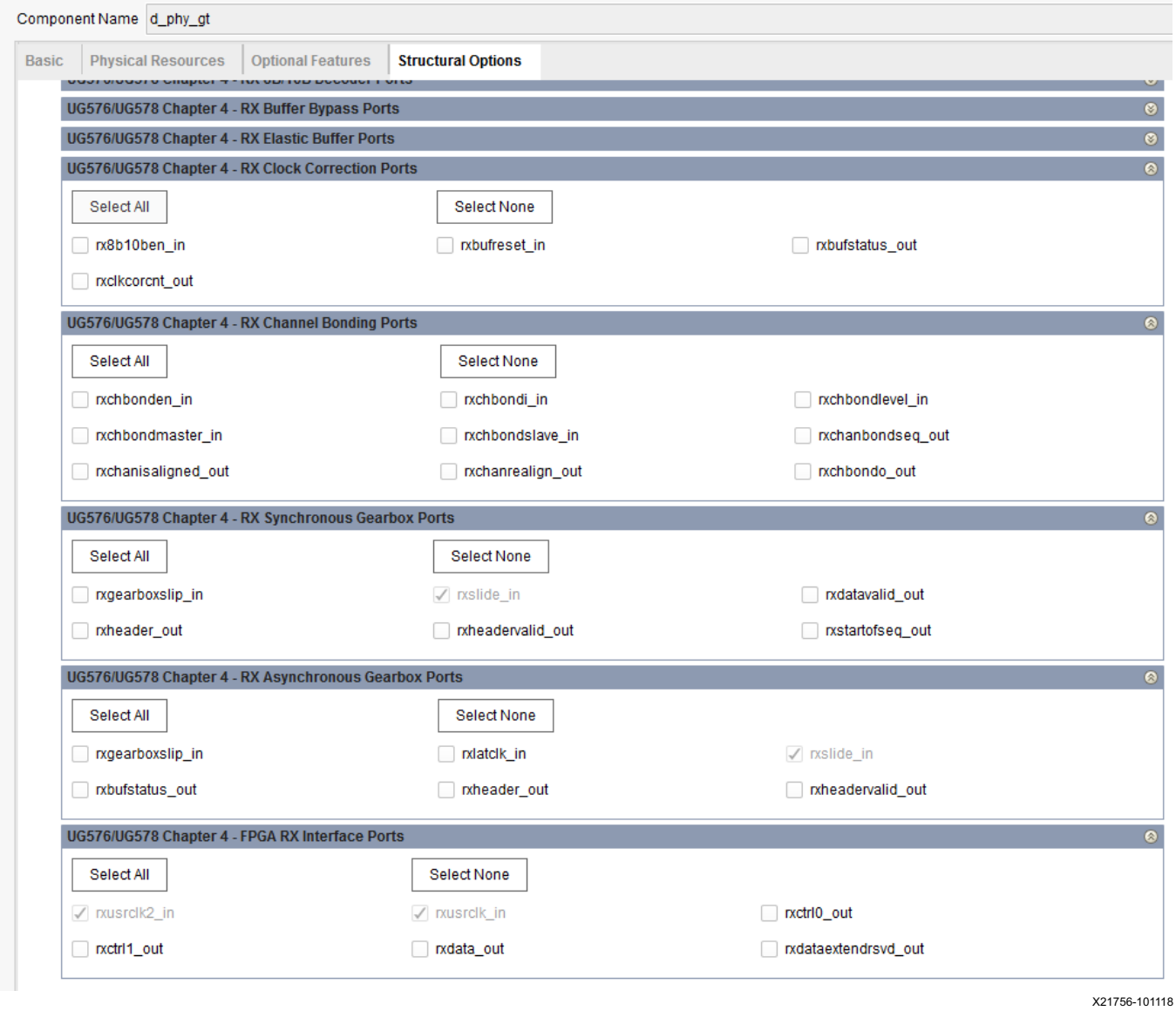


Figure 44: **Chapter 4 - RX Clock Correction Ports Section**

2.5 Gb/s GTH D-PHY Custom Design and GTH Transceiver Migration Guidelines

To customize the MIPI design (number of MIPI channels or GTH placement) based on this reference design, read the following carefully before modifying the code.

GTH Reference Clock Configuration

USER TOP (KCU105 loop TOP) contains the code shown in Figure 45. The following configuration is according to the clock solution of the example design of the KCU105 demo board. Customers need to modify the parameters based on the location of the GTH transceiver on the board [Ref 2].

```
gth_phy_top  #
 (
   .SIMULATION                    ( SIMULATION                )
  ,.DEVICE                        ( DEVICE                    )
) gth_phy_top_u
 (
   .sys_rst                       ( sys_rst                   )
  ,.qpll0refclksel_in             ( {{3'b110}, {3'b001}}      )
  ,.qpll1refclksel_in             ( {{3'b110}, {3'b001}}      )
  ,.cpllrefclksel_in              ( {{4{3'b101}}, {4{3'b010}}} )
  ,.q0_refclk0                    ( rx_refclk                 )
  ,.q0_refclk1                    ( tx_refclk                 )
  ,.q0_north_refclk0              ( 1'b0                      )
  ,.q0_north_refclk1              ( 1'b0                      )
  ,.q0_south_refclk0              ( 1'b0                      )
  ,.q0_south_refclk1              ( 1'b0                      )
  ,.q1_refclk0                    ( q1_refclk0                )
  ,.q1_refclk1                    ( q1_refclk1                )
  ,.q1_north_refclk0              ( 1'b0                      )
  ,.q1_north_refclk1              ( 1'b0                      )
  ,.q1_south_refclk0              ( rx_refclk                 )
  ,.q1_south_refclk1              ( tx_refclk                 )
  ,.gthrxn_in                     ( gthrxn_in                 )
  ,.gthrxp_in                     ( gthrxp_in                 )
  ,.gthtxn_out                    ( gthtxn_out                )
  ,.gthtxp_out                    ( gthtxp_out                )
  ,.freerun_clk                   ( freerun_clk               )
  ,.mipi_recover_clk              (                           )
  ,.txpllclksel_in                ( txpllclksel_in            )
```

X21684-100418

Figure 45: **GTH Reference Clock Configuration in HDL Code**

RX/TX 5lane Module

To support a wide range of design requirements, the reference design provides a fixed 4 data lane + 1 clock lane MIPI TX and RX data processing module connected to the MIPI D-PHY PCS layer. Each transceiver module maps to five standard GTH lanes named the RX/TX 5lane module. This module is provided in the netlist.

You can generate GTH modules according to the method described in Implementing the 2.5G Transceiver D-PHY Reference Design, customize the number of MIPI interface data lanes (data lane < 4), and connect RX/TX 5lane modules through the logical channel adaptation layer. When the number of data lanes is less than 4, the Vivado tools compile automatically to optimize those channels by processing unused data channel ports on RX/TX 5lane.

Logical Channel Adaptation Layer between MIPI TX/RX5 Lane Module and GTH Transceiver

The data lane of the MIPI interface is sorted from D0 to DN. The RX/TX 5lane's data lane is also sorted in this lane order, mapping one-to-one with the MIPI interface data.

To simplify the mapping relationship between the RX/TX 5lane data and clock lanes and the transceiver physical lane, and to separate the transmitter and receiver channels from the GTH transceiver (which must include both transmitter and receiver channels when the GTH transceiver IP is generated), some HDL code is added between `GTH_PHY_TOP.v` and `GTH_transceiver_example_wrapper.v` to implement a logical channel adaptation layer.

In this adapter layer, N (N < 8) independent logical channels are set up, and these N logical channels are divided into two logical Quads, which are numbered in HDL code. Logical channels 0–3 (Q0 0–3) are included by logical Q0. Logical channels 4–7 (Q1 0–3) are contained by logical Q1. The logical Quad has a mapping relationship with the physical transceiver Quad in code, and the logical channel also has a mapping relationship with the physical transceiver channel. In the HDL code, an example of a logical Quad is shown in Figure 46.

```
gth_phy_top  #
 (
   .SIMULATION                    ( SIMULATION                )
  ,.DEVICE                        ( DEVICE                    )
) gth_phy_top_u
 (
   .sys_rst                       ( sys_rst                   )
  ,.qpll0refclksel_in             ( {{3'b110}, {3'b001}}      )
  ,.qpll1refclksel_in             ( {{3'b110}, {3'b001}}      )
  ,.cpllrefclksel_in              ( {{4{3'b101}}, {4{3'b010}}} )
  ,.q0_refclk0                    ( rx_refclk                 )
  ,.q0_refclk1                    ( tx_refclk                 )
  ,.q0_north_refclk0              ( 1'b0                      )
  ,.q0_north_refclk1              ( 1'b0                      )
  ,.q0_south_refclk0              ( 1'b0                      )
  ,.q0_south_refclk1              ( 1'b0                      )
  ,.q1_refclk0                    ( q1_refclk0                )
  ,.q1_refclk1                    ( q1_refclk1                )
  ,.q1_north_refclk0              ( 1'b0                      )
  ,.q1_north_refclk1              ( 1'b0                      )
  ,.q1_south_refclk0              ( rx_refclk                 )
  ,.q1_south_refclk1              ( tx_refclk                 )
  ,.gthrxn_in                     ( gthrxn_in                 )
  ,.gthrxp_in                     ( gthrxp_in                 )
  ,.gthtxn_out                    ( gthtxn_out                )
  ,.gthtxp_out                    ( gthtxp_out                )
  ,.freerun_clk                   ( freerun_clk               )
  ,.mipi_recover_clk              (                           )
  ,.txpllclksel_in                ( txpllclksel_in            )
```

X21685-100418

Figure 46: **Logical Quad in HDL Code**

Note: Between the external MIPI interface data lane N to 0 and FPGA internal MIPI TX/RX 5lane modules data lane N to 0, the data lane has a strict one-to-one correspondence from N to 0.

Figure 47 describes the logical architecture of the channel adaptation layer in the reference design.

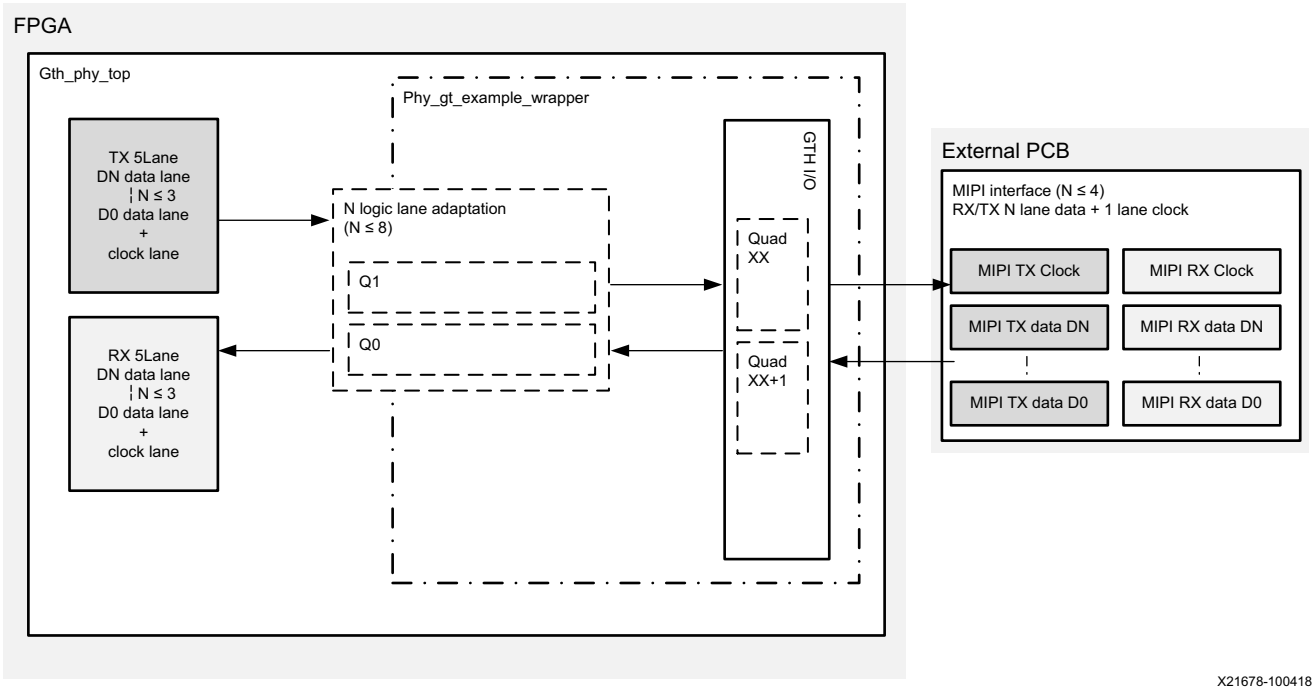


Figure 47: **Logical Architecture of Channel Adaptation Layer in Reference Design**

On the GTH transceiver side, depending on the number of MIPI interface data lanes on the PCB and design requirements, the physical location of the GTH transceiver can first be assigned, and then the XDC constraints can be used to map between the GTH transceiver and logical channels. See the constraint in Figure 49. The MIPI TX/RX 5lane module lane D0 to DN and MIPI interface D0 to DN are re-mapped by logical channel adaptation layer within the FPGA.

The behavior of the MIPI data and clock lanes is equivalent to the logic channel adaptation layer. However, the MIPI data lane has multiple lanes and a fixed lane number corresponding to the 5lane module data lane, so usually, the MIPI data lane is constrained first, followed by the MIPI clock lane on the GTH transceiver.

After determining the GTH transceiver position, the transceiver Quad location corresponding to logical Q needs to be determined to configure the reference clock source. Figure 48 is a mapping relation taking the KCU105 reference design as an example. (For GTH mapping refer to Figure 24.) The figure shows MIPI TX 4 data lane + 1 clock lane, and MIPI RX 4 data lane + 1 clock lane.

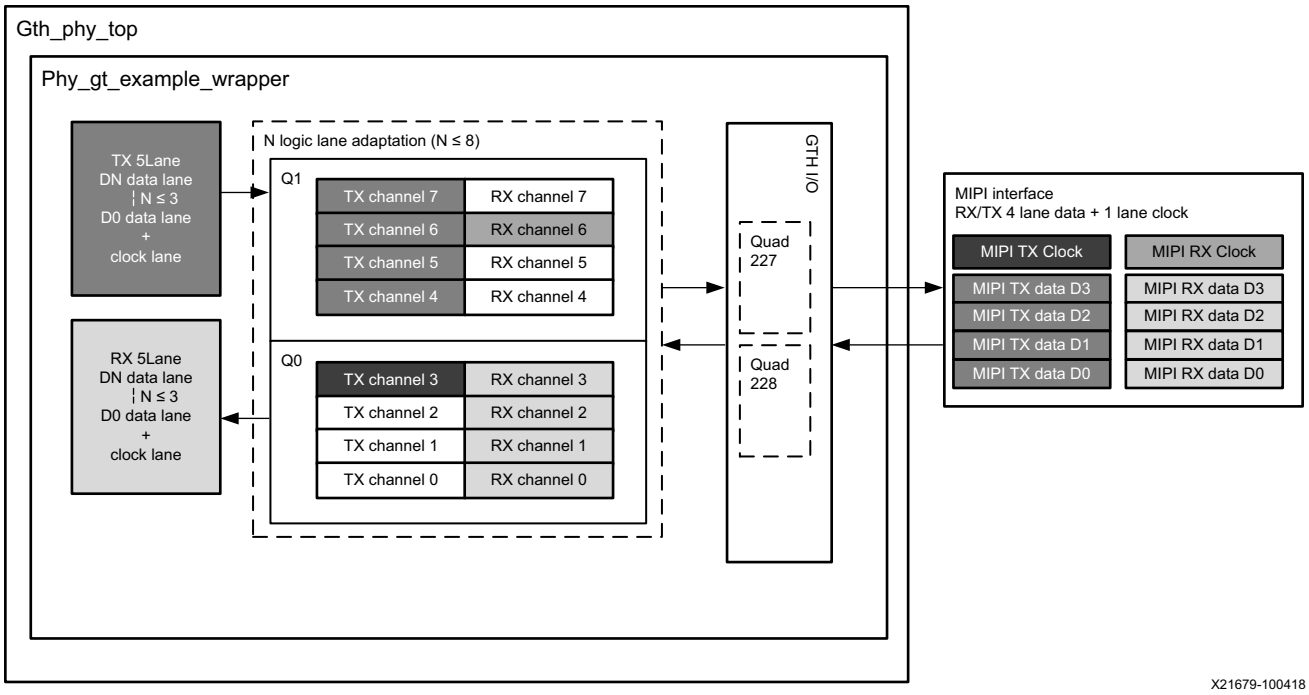


Figure 48: **Mapping Relation of KCU105 Reference Design**

1. The MIPI interface TX is assigned to 4X GTH of a transceiver Quad, and RX4 data lane is assigned to other 4X GTH of a transceiver Quad. In the reference design, the MIPI RX interface is connected to logical channels 0–3 (Q0 0–3) of the logical Q0, and the MIPI interface TX is connected to the Q0 logical channels 4–7 (Q1 0–3). On the other side of the logical channel, to correctly connect the MIPI interface data lane D0–D3 to the TX/RX 5lane data lane D0–D3, the RX 5lane module data lane connects to logical channels 0–3 (Q0–3), and the TX 5lane module data lane connects to logical channels 4–7 (Q1 0–3).
2. In the reference design, the MIPI RX clock lane and MIPI TX data lane share a GTH transceiver, while the MIPI TX clock lane and MIPI RX data lane share a GTH transceiver. Because the data lane determines the logical channel for the transceiver, the clock lane has the same logical channel number as the shared GTH data lane. The MIPI RX clock in the reference design is projected to logical lane 6 (Q1 2), so the clock in the RX 5lane module needs to be connected to logical lane 6. TX clocks are projected to logical lane 3 (Q0 3), and then the clock lane of the TX 5lane module needs to be connected to logical lane 3 (adapted according to the KCU105 demo board design).
3. Modify the MIPI_TX_USED and MIPI_RX_USED parameters in `GTH_PHY_TOP.v` to enable the corresponding DRP CH. In the KCU105 reference design:

   MIPI_TX_USED = `8'b11111000`

   MIPI_RX_USED = `8'b01001111`

4. In the KCU105 reference design, logic Q0 corresponds to Quad228, and Q1 corresponds to Quad227. If the reference clock is output from Quad228, connect the QPLL/CPLL on each GTH Quad to Refreclk0/Refreclk1 and SouthRef0/SouthRef1 in the HDL code.

5. Add constraints in the XDC to implement MIPI data lane (the MIPI interface connects to the GTH transceiver) and FPGA logic channel mapping.

```
# KCU105 demo
#// KC105 define
#//GT LANE MAPPING Q227
#// GT Locataion    Logic
#// LOC=X0Y14       RX lane7  ----  NC
#// LOC=X0Y13       RX lane6  ----  MIPI RX CLK  RX MIPI CLOCK LANE
#// LOC=X0Y12       RX lane5  ----  NC
#// LOC=X0Y15       RX lane4  ----  NC
#//GT LANE MAPPING Q228
#// LOC=X0Y19       RX lane3  ----  MIPI RX D3   RX MIPI DATA LANE
#// LOC=X0Y18       RX lane2  ----  MIPI RX D2   RX MIPI DATA LANE
#// LOC=X0Y17       RX lane1  ----  MIPI RX D1   RX MIPI DATA LANE
#// LOC=X0Y16       RX lane0  ----  MIPI RX D0   RX MIPI DATA LANE/MASTER
#//GT LANE MAPPING Q227
#// LOC=X0Y14       TX lane7  ----  MIPI TX D3   TX MIPI DATA LANE
#// LOC=X0Y13       TX lane6  ----  MIPI TX D2   TX MIPI DATA LANE
#// LOC=X0Y12       TX lane5  ----  MIPI TX D1   TX MIPI DATA LANE
#// LOC=X0Y15       TX lane4  ----  MIPI TX D0   TX MIPI DATA LANE/MASTER
#//GT LANE MAPPING Q228
#// LOC=X0Y19       TX lane3  ----  MIPI TX CLK  TX MIPI CLOCK LANE
#// LOC=X0Y18       TX lane2  ----  NC
#// LOC=X0Y17       TX lane1  ----  NC
#// LOC=X0Y16       TX lane0  ----  NC
set_property LOC GTHE3_CHANNEL_X0Y16 [get_cells -hierarchical -filter {NAME =~ *gen_channel_container[3].*gen_gthe3_channel_inst[0].GTHE3_CHANNEL_PRIM_INST}]
set_property LOC GTHE3_CHANNEL_X0Y17 [get_cells -hierarchical -filter {NAME =~ *gen_channel_container[3].*gen_gthe3_channel_inst[1].GTHE3_CHANNEL_PRIM_INST}]
set_property LOC GTHE3_CHANNEL_X0Y18 [get_cells -hierarchical -filter {NAME =~ *gen_channel_container[3].*gen_gthe3_channel_inst[2].GTHE3_CHANNEL_PRIM_INST}]
set_property LOC GTHE3_CHANNEL_X0Y19 [get_cells -hierarchical -filter {NAME =~ *gen_channel_container[3].*gen_gthe3_channel_inst[3].GTHE3_CHANNEL_PRIM_INST}]
set_property LOC GTHE3_CHANNEL_X0Y15 [get_cells -hierarchical -filter {NAME =~ *gen_channel_container[4].*gen_gthe3_channel_inst[0].GTHE3_CHANNEL_PRIM_INST}]
set_property LOC GTHE3_CHANNEL_X0Y12 [get_cells -hierarchical -filter {NAME =~ *gen_channel_container[4].*gen_gthe3_channel_inst[1].GTHE3_CHANNEL_PRIM_INST}]
set_property LOC GTHE3_CHANNEL_X0Y13 [get_cells -hierarchical -filter {NAME =~ *gen_channel_container[4].*gen_gthe3_channel_inst[2].GTHE3_CHANNEL_PRIM_INST}]
set_property LOC GTHE3_CHANNEL_X0Y14 [get_cells -hierarchical -filter {NAME =~ *gen_channel_container[4].*gen_gthe3_channel_inst[3].GTHE3_CHANNEL_PRIM_INST}]

set_property LOC GTHE3_COMMON_X0Y4 [get_cells -hierarchical -filter {NAME =~ *gth_com_0/common_inst/gthe3_common_gen.GTHE3_COMMON_PRIM_INST}]
set_property LOC GTHE3_COMMON_X0Y3 [get_cells -hierarchical -filter {NAME =~ *gth_com_1/common_inst/gthe3_common_gen.GTHE3_COMMON_PRIM_INST}]
```

X21686-100418

Figure 49: **GTH Mapping to PCB**

Based on this mapping method, customers can customize GTH mapping by modifying the source code (Figure 50). For example, you can place four MIPI TX data lanes on the GTH transceiver channel shared by the transceiver Quad where the MIPI RX data lane is located. The MIPI TX clock lane is placed in the MIPI RX clock lane to share a GTH transceiver channel. The DHL code needs to be modified in the reference design to connect the TX 5lane module data lane to TX logical channels 0–3 (Q0–3), TX 5lane module clock lane to TX logical channel 6 (Q1 2), and RX 5Lane needs no change. The corresponding DRP module enabling parameters and related reset chains should be modified as shown below:

MIPI_TX_USED = `8'b01001111`

MIPI_RX_USED = `8'b01001111`

Update the XDC with the new TX GTH location. At the same time, the location of the transceiver Quad has changed, and the clock parameters of the logical Q0 and Q1 in the code must be modified accordingly.

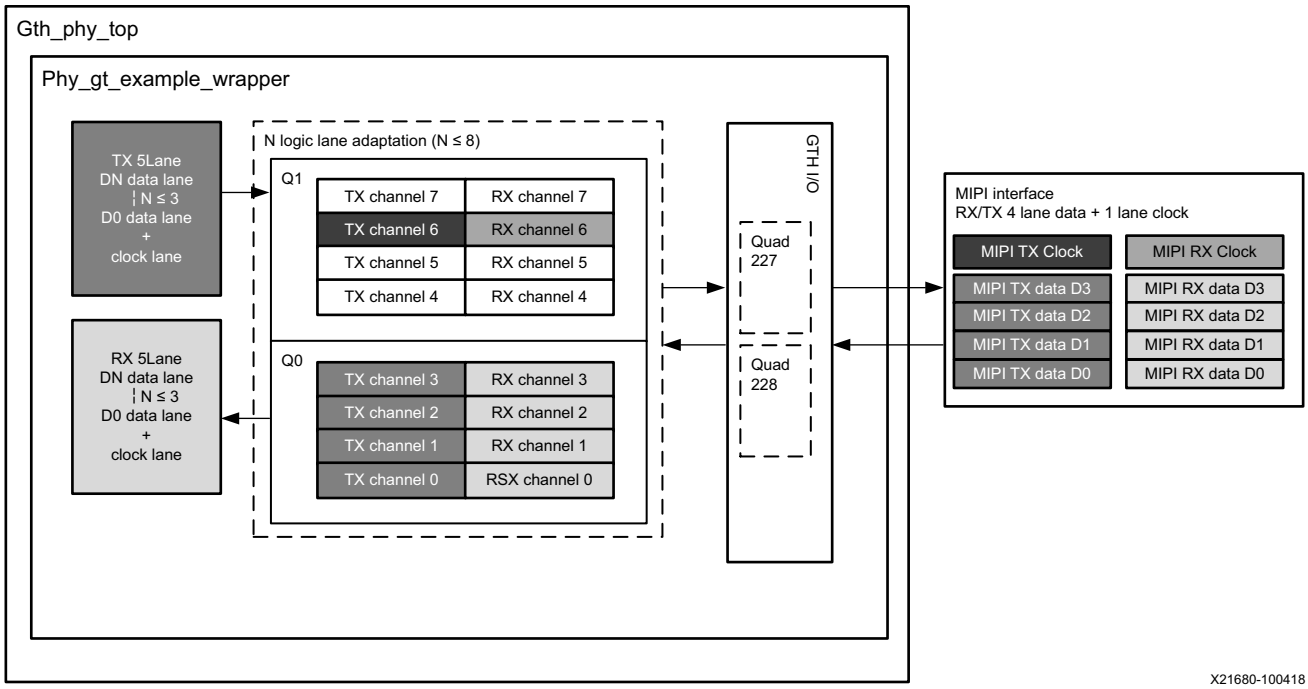


Figure 50: **Mapping Relation of a Customize Design Example**

GTH DRP Chain

Referring to Figure 51, the GTH PHY TOP contains N (N < 8) DRP channel control units and two DRP common control units to support a multiple line rate MIPI interface. In the RX 5Lane module, there is a DRP control module for CDR five times oversampling. All the design requirements of the DRP control module can be referred to *UltraScale Architecture GTH Transceivers User Guide* (UG576) [Ref 2].

IMPORTANT: *If you redefine the MIPI channel number and change the GTH transceiver placement based on the reference design, you also need to modify the DRP chain.*

Note: The DRP channel module is optional in MIPI fixed-line rate mode, but the DRP common module is required.

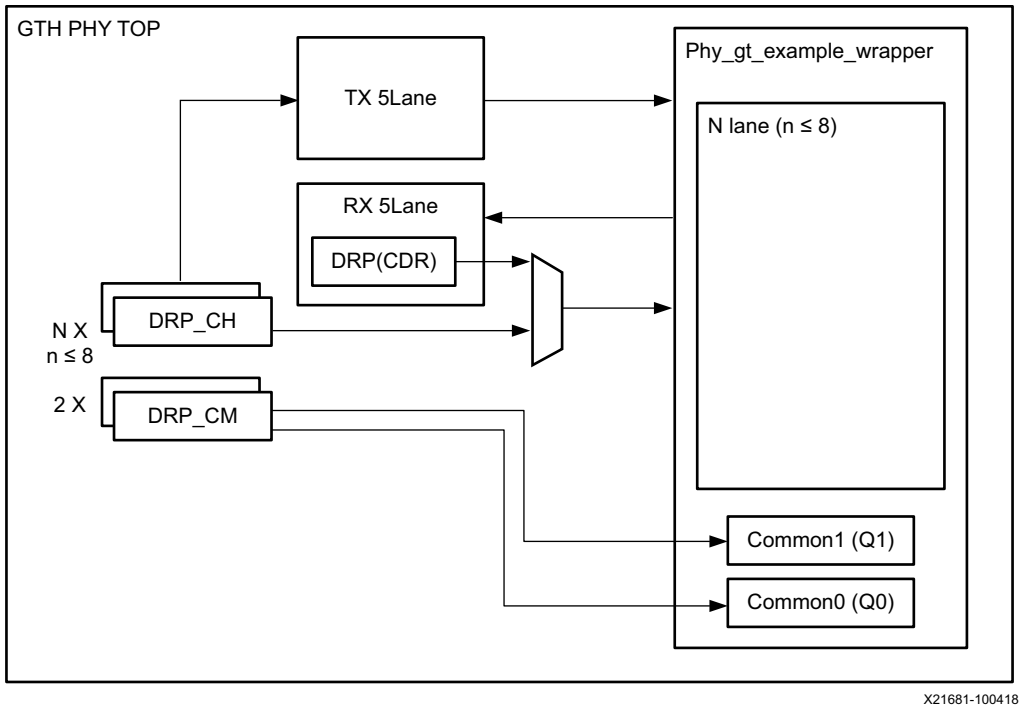


Figure 51: **DRP Chain of 2.5G MIPI D-PHY Core**

IMPORTANT: *All the source code in the reference design can be modified, except the RX/TX 5lane module. In theory, the code can be modified to change the location of the GTH transceiver and the channel number. At present, only designs with data lane ≤ 4 are supported. Follow the requirements in the UltraScale Architecture GTH Transceivers User Guide (UG576) [Ref 2] and comply with the reference design constraints. Contact the local Xilinx sales office for support.*

IMPORTANT: *If you modify the TX/RX lane, all the GTH clock chains, reset chains, control paths, and DRP chains should be modified together.*

Summary of GTH Transceiver Mapping

Xilinx recommends carrying out the GTH transceiver mapping according to the reference design (Figure 24). Use the same GTH transceiver location relationship as the reference design on your board so that your new design is easier to implement, with only a few lines of code about the clock needing to be modified.

Observe the following guidelines to design a custom MIPI interface and modify the GTH transceiver mapping:

1. According to the rules described in *UltraScale Architecture GTH Transceivers User Guide* (UG576) [Ref 2] and I/O Assign/Timing Constraints of the 2.5G MIPI Solution, the GTH clock should be determined before GTH placement, making sure to allocate the CPLL and QPLL clock resources.
2. The reference design is a TX/RX 4 data lane + 1 clock lane design. If you have different lane and location requirements, refer to Implementing the 2.5G Transceiver D-PHY Reference

Design to regenerate the GTH IP based on the reference design, then modify the HDL code to connect to the TX/RX 5lane module.

3. The modules related to GTH mapping provide customers with open source code, namely, the User_top (`KCU105_loop_top.v`), `Gth_Phy_top.v`, and Phy_gt_example_wrapper modules. Users need to modify the GTH port mapping, reset chain, and DRP interface according to the reference design.

Multi-Protocol Migration

The 2.5G transceiver D-PHY solution is flexible in the selection of QPLL or CPLL. At the same time, the reference design can flexibly extract all the transceiver ports that are not used at the MIPI level. Thus, it is possible to migrate the 2.5G transceiver D-PHY design to the same eight lanes as the transceiver interface of other protocols. Attention should be paid to the following:

- The key point is how to design a clock system, such as in Figure 52.

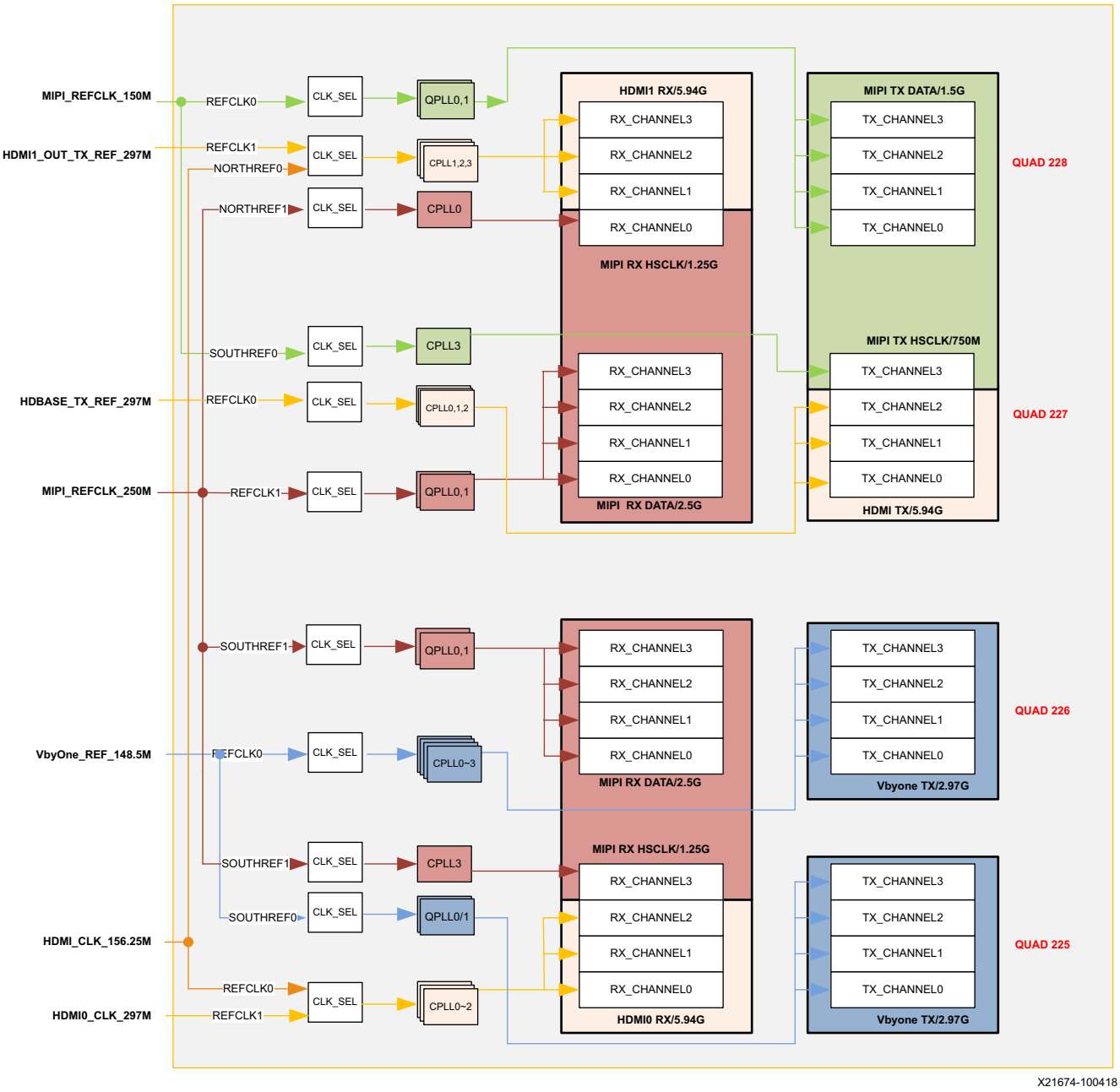


Figure 52: **A Complex Customer Design Transceiver Mapping and GTH Transceiver Clock Solution Example (UltraScale FPGAs)**

- Define the use of the DRP interface, and understand the DRP control chain of the reference design. Whether or not other protocol transceiver interfaces need to use DRP, such as when using DRP, you must consider how to time-share DRP ports with MIPI interfaces.
- According to the designed clock system, modifying the open HDL code and the reset chain design is very critical.

Porting the Design from UltraScale to UltraScale+ FPGAs

To port your design from UltraScale to UltraScale+ FPGAs, use an UltraScale+ FPGA D-PHY reference design. Hardware design rules fully comply with the UltraScale FPGA design requirements.

Note: The current UltraScale+ FPGA GTH transceiver clock system is more complex, and the UltraScale+ FPGA 2.5G MIPI D-PHY reference design currently does not support CPLL. This limitation must be taken into account when designing the GTH transceiver mapping to previous PCB designs.

PCB Design Guidelines

1. Component placement:

   The attenuation network resistors on the TX side must be close to the FET switch device, but the attenuation network resistors (see Figure 10) and the FET switch device can be placed at any location on the PCB.

2. Guidelines and recommendations:

   For the PCB wiring of the HS section, refer to the requirements in the "Design of Transitions for High-Speed Signals" chapter of *UltraScale Architecture PCB Design User Guide* (UG583) [Ref 8] for optimizing the impedance continuity of the vias and pads.

   There is no special requirement for the LP device.

3. Guidelines for insertion loss:

   The GTH transceiver-based D-PHY supports TX pre-emphasis and RX equalization functions. TX pre-emphasis can support 0~6 dB insertion loss by setting the TXPOSTCURSOR[4:0] for line rates higher than 500 Mb/s. RX equalization functions can support 0~6 dB insertion loss by setting these attributes. (This is included in the reference design.)

   For UltraScale FPGAs:

   - RXDFELPM_KL_CFG0[15:9] = 0x40;
   - RXLPM_KH_CFG0[15:9] = 0x40;
   - RXLPM_GC_CFG[12:8] = 0x1F;

   For UltraScale+ FPGAs:

   - RXDFELPM_KL_CFG1[15:9] = 0x40;
   - RXLPM_KH_CFG1[15:9] = 0x40;
   - RXLPM_GC_CFG[15:9] = 0x1F;

   Figure 53 shows eye diagrams of a data stream after 6 dB insertion loss (left) and after being processed by equalization circuits (right).

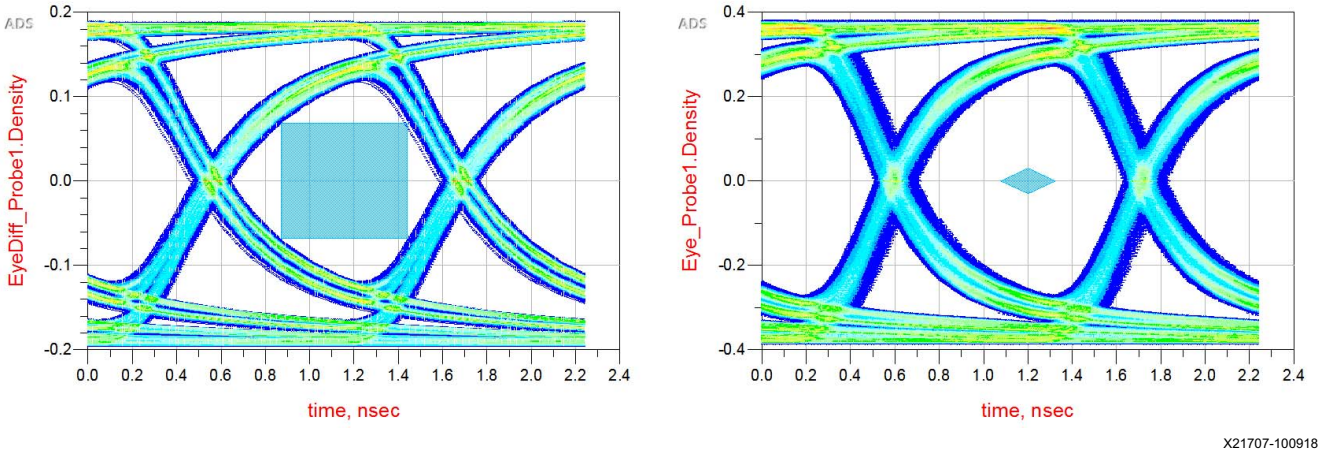


Figure 53: **Data Stream Eye Diagrams**

Reference Design

Download the reference design files for this application note from the Xilinx website.

Table 17 shows the reference design matrix.

Table 17: **Reference Design Matrix**

Parameter	Description
General	
Developer name	John Hu
Target devices	UltraScale and UltraScale+ FPGAs
Source code provided	Yes
Source code format	
Design uses code and IP from existing Xilinx application note and reference designs or third party	MIPI CSI-2 RX and MIPI DSI-2 TX IP provided by Northwest Logic
Simulation	
Functional simulation performed	No
Timing simulation performed	N/A
Test bench used for functional and timing simulations	No
Test bench format	Verilog
Simulator software/version used	Vivado tools 2018.2
SPICE/IBIS simulations	N/A
Implementation	
Synthesis software tools/versions used	Vivado tools 2018.2

Table 17: **Reference Design Matrix *(Cont'd)***

Parameter	Description
Implementation software tools/versions used	Vivado tools 2018.2
Static timing analysis performed	No
Hardware Verification	
Hardware verified	Yes
Hardware platform used for verification	KCU105 evaluation board

2.5G MIPI D-PHY Design (Only for Simulation)

Figure 54 shows the HDL hierarchy in the 2.5G MIPI D-PHY design. `Prbs_any.v` is used to generate and detect the test signal and is not necessary in the customer design. The 2.5G MIPI D-PHY design contains simulation files that support the simulation of MIPI HS signals.

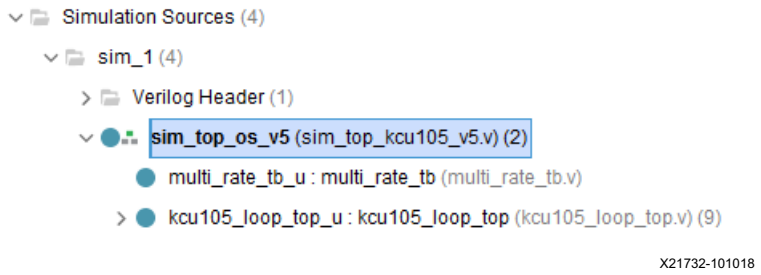


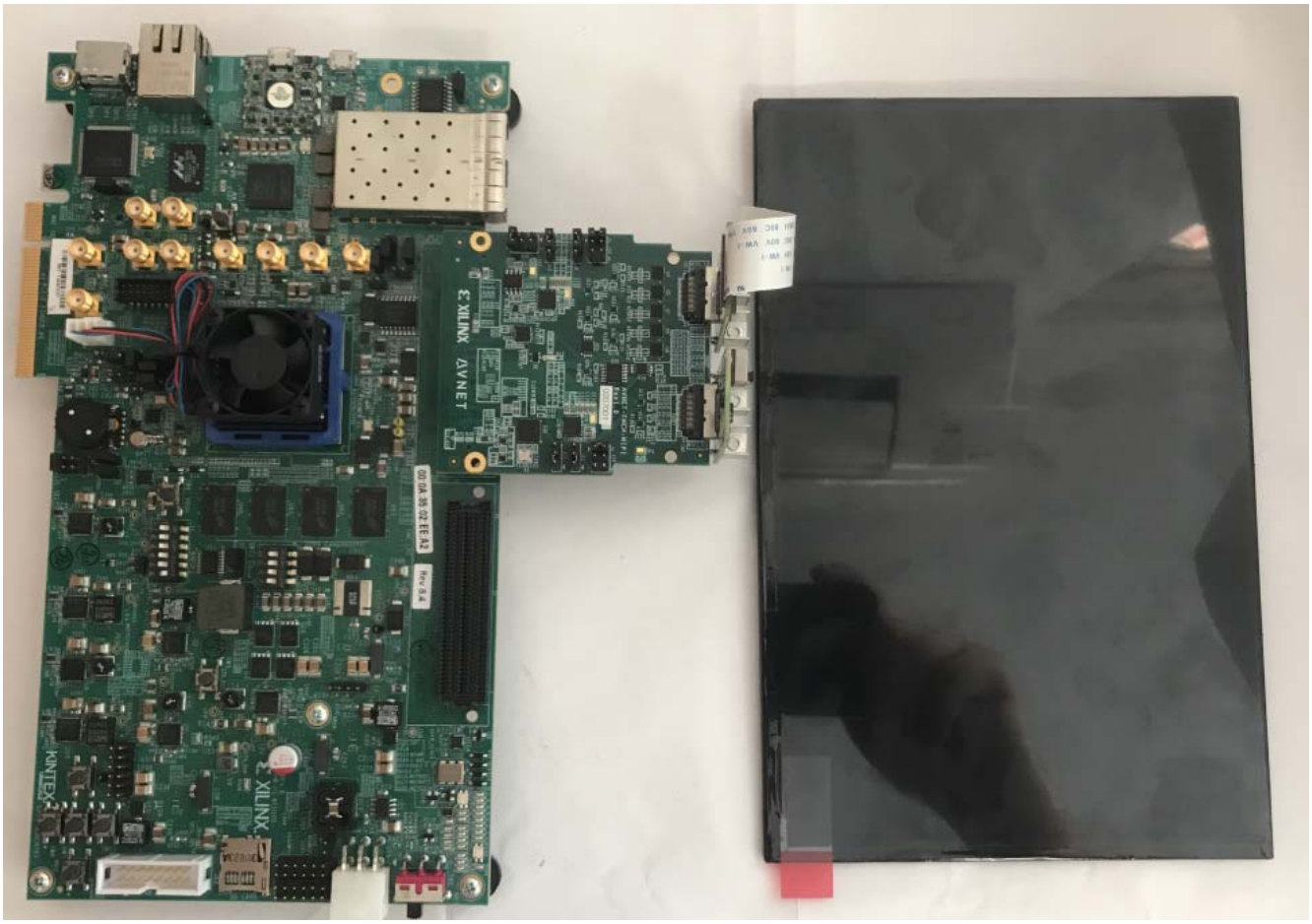
Figure 54: **HDL Hierarchy in 2.5G MIPI D-PHY Design**

Note: LP signal simulation is not supported by the 2.5G MIPI D-PHY design because the LP signal contains a handshake protocol and is managed by the MIPI MAC layer. You can integrate the 2.5G D-PHY design with MIPI CSI-2/MIPI DSI-2 IP to develop a demonstrable system on this hardware platform.

The following two subsystem demonstrations use these MIPI CSI-2 RX/MIPI DSI-2 TX IP cores from Xilinx IP partner Northwest Logic.

Hardware Platform

To build a 2.5G MIPI demonstration platform, you require a KCU105 board [Ref 9] or ZCU102 board [Ref 10] with an Avnet 2.5G MIPI development suite (card number: ADX=FMC-MIPI2.5G). shows an overview of the hardware platform.



X21710-100918

Figure 55: **Hardware Platform Overview**

The Avnet 2.5G MIPI development suite contains several sub-cards:

- Avnet 2.5G MIPI FMC card (Figure 55).

X21711-100918

Figure 56: **Avnet 2.5G MIPI FMC Card**

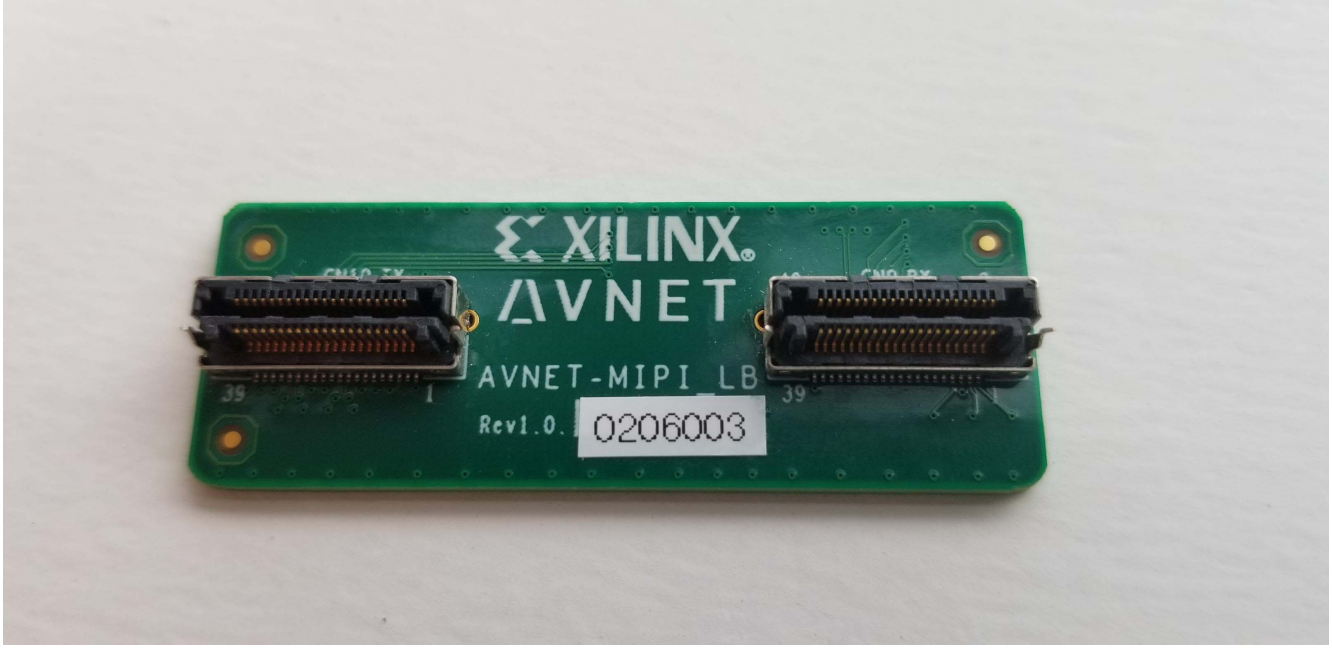
Note: MIPI TX and RX connectors are included on this board. Each connector supports connections to MIPI CSI-2 or MIPI DSI-2 interfaces.

- MIPI camera display adapter cards (Figure 57).

X21713-101018

Figure 57: **MIPI Camera Adapter Card (Left) and Display Adapter Card (Right)**

- MIPI loopback adapter card (Figure 58).



X21733-101018

Figure 58: **MIPI Loopback Adapter Card**

2.5G MIPI Subsystem in Pass-through Mode

Figure 59 shows the 2.5G MIPI subsystem in pass-through mode.

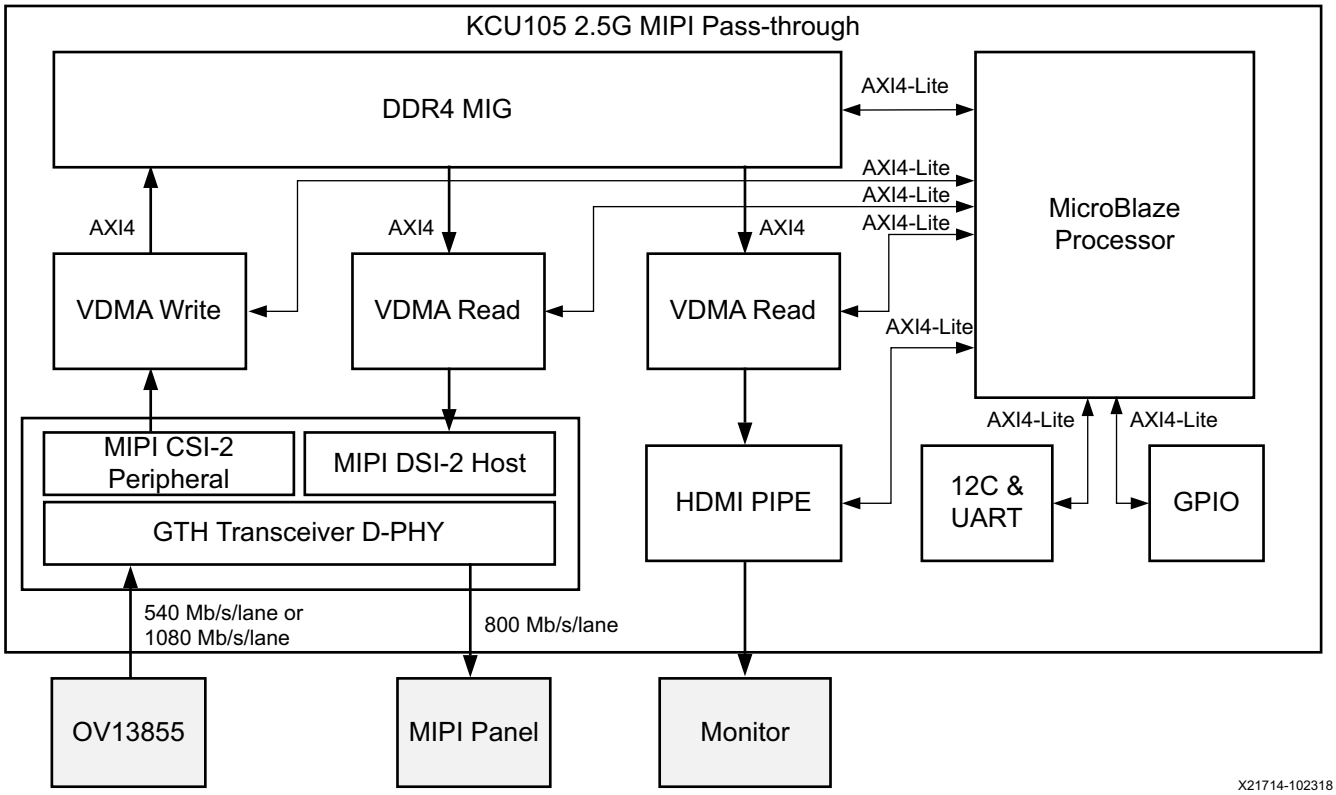


Figure 59: **2.5G MIPI Subsystem in Pass-through Mode**

To run this demo, you require the KCU105 board, Avnet 2.5G MIPI FMC card, Avnet MIPI camera adapter card, and MIPI display adapter card. The video stream generated by the MIPI camera adapter card is sent to the MIPI DSI-2 TX (on the Avnet 2.5G MIPI FMC card) and HDMI™ interface (on the KCU105 board) simultaneously through the MIPI CSI-2 RX interface (on the Avnet 2.5G MIPI FMC card). The DSI-2 TX can also be switched to send a fixed video pattern from the Test Pattern Generator module.

Note: For more information about how to set up the demo, see the `readme.txt` in the reference design ZIP file.

Running the 2.5G MIPI Subsystem Demo in Pass-through Mode on the KCU105 Board

Xilinx only provides the bit file for the demo (limited by the Northwest Logic IP license agreement). You can build the demo quickly using the hardware platform and programming the FPGA on the KCU105 board with the bit file.

Note: For more information about the 2.5G MIPI subsystem hardware and logic design, contact Avnet and Northwest Logic.

2.5G Subsystem Design in Loopback Mode

Figure 59 shows the 2.5G MIPI subsystem in loopback mode.

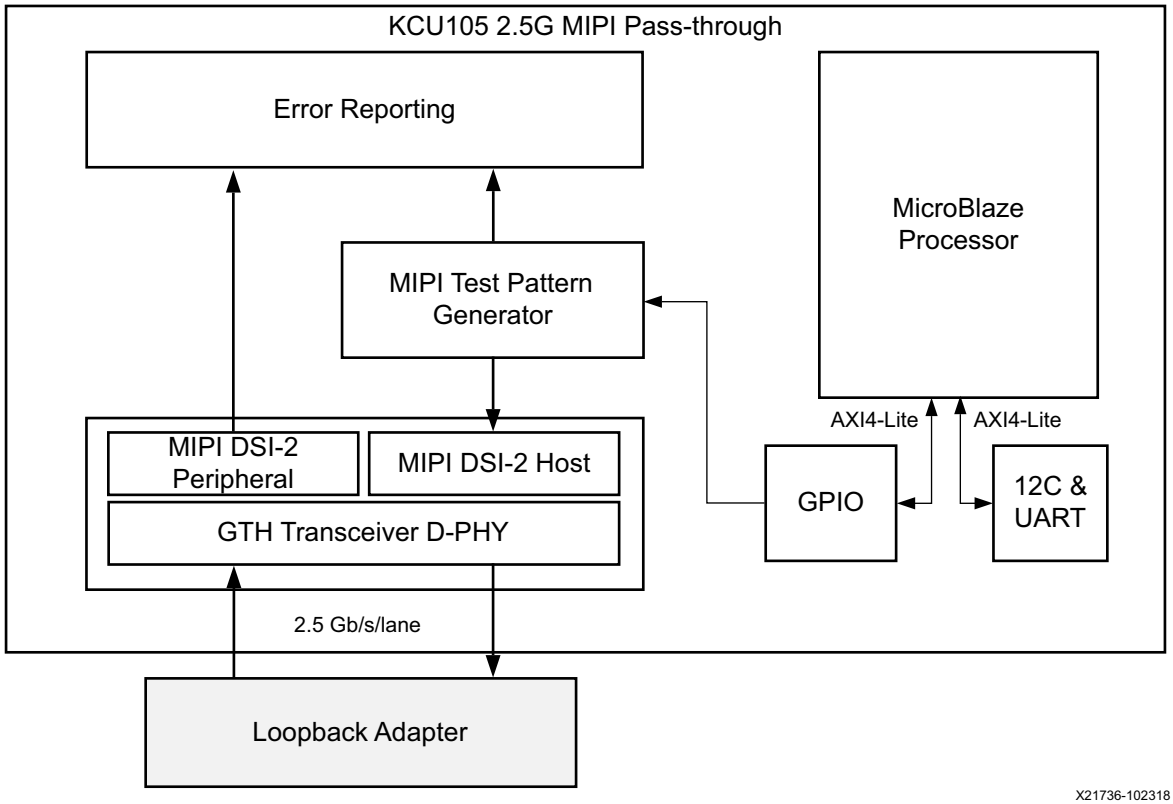


Figure 60: **2.5G MIPI Subsystem in Loopback Mode**

To run this demo, you require the KCU105 board, Avnet 2.5G MIPI FMC card, and MIPI loopback adapter. It is currently difficult to find a 2.5G MIPI interface camera in the market, so the loopback card was provided by the Avnet 2.5G MIPI development suite. This was used to connect the MIPI RX and DSI TX on the FMC board directly for the 2.5G MIPI loopback test.

Note: For more information about how to set the demo, read the `readme.txt` in the reference design ZIP file.

Running the 2.5G MIPI Subsystem in Loopback Mode on the KCU105 Board

Xilinx only provides the bit file for the demo (limited by the Northwest Logic license agreement). You can build the demo quickly using the hardware platform and programming the FPGA on the KCU105 board with the bit file.

Note: For more information about the 2.5G MIPI subsystem hardware and logic design, contact Avnet and Northwest Logic.

KCU105 Board FMCH VADJ Adjustment

The 2.5G MIPI FMC's intelligent platform management interface (IPMI) can make the KCU105 board adjust the VDJ automatically so that users do not need to make any adjustments.

Conclusion

The use of simple external circuits, such as those presented in this application note, make it possible to connect an FPGA to an ASSP device via its MIPI interface, supporting up to 2.5G.

Documentation Navigator and Design Hubs

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

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

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

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

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

References

1. MIPI Alliance specification for D-PHY
   http://mipi.org/specifications/physical-layer
2. *UltraScale Architecture GTH Transceivers User Guide* (UG576)
3. *MIPI D-PHY v4.1 Product Guide* (PG202)
4. *D-PHY Solutions* (XAPP894)
5. Scalable Low-Voltage Signaling for 400 mV: JESD 8-13
   https://www.jedec.org/standards-documents/docs/jesd-8-13
6. *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics* (DS892)
7. *Kintex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics* (DS922)
8. *UltraScale Architecture PCB Design User Guide* (UG583)
9. Xilinx Kintex UltraScale FPGA KCU105 Evaluation Kit
   https://www.xilinx.com/products/boards-and-kits/kcu105.html

10. Xilinx Zynq UltraScale+ MPSoC ZCU102 Evaluation Kit
https://www.xilinx.com/products/boards-and-kits/ek-u1-zcu102-g.html

11. Avnet ADX=FMC-MIPI2.5G sub-card
https://www.avnet.com/wps/portal/apac

Revision History

The following table shows the revision history for this document.

Section	Revision Summary
10/31/2018 Version 1.0	
Initial Xilinx release.	N/A

Please Read: Important Legal Notices