# Virtex UltraScale+ FPGAs GTM Transceivers

## *User Guide*

**UG581 (v1.3) May 21, 2020**

XILINX®

# Revision History

The following table shows the revision history for this document.

| Section | Revision Summary |
|---|---|
| **05/21/2020 Version 1.3** | |
| Features | Updated minimum PAM4 line rate to 20.6 Gb/s and NRZ line rate to 10.3 Gb/s. |
| Figure 1: GTM Dual Configuration | Updated direction of signals into and out of REFCLK distribution. |
| Reference Clock Selection and Distribution | Updated NRZ line rate to 14.5 Gb/s and PAM4 line rate to 29 Gb/s. |
| LCPLL | Updated NRZ line rate to 28.21 Gb/s and PAM4 line rate to 56.42 Gb/s. |
| Loopback | Updated near-end PMA loopback bullet. |
| Using the TX Buffer | Added new topic. |
| Table 45: TX Configurable Driver Ports | Updated values for CH[0/1]_TXDRVAMP[4:0], CH[0/1]_TXEMPPRE[4:0], CH[0/1]_TXEMPPRE2[3:0], and CH[0/1]_TXEMPPOST[4:0]. |
| Using RX Pattern Checker | Updated steps to disable RX pattern checker for 80-bit or 160-bit data widths. |
| **09/20/2019 Version 1.2** | |
| Table 45: TX Configurable Driver Ports | Updated descriptions of CH[0/1]_TXDRVAMP[4:0], CH[0/1]_TXEMPMAIN[5:0], CH[0/1]_TXEMPPRE[4:0], CH[0/1]_TXEMPPRE2[3:0], and CH[0/1]_TXEMPPOST[4:0]. |
| Loopback | Added instructions on how to properly enable PCS near-end loopback. |
| RX CDR Lock to Reference | Added new topic. |
| RX Margin Analysis | Updated description of sample eye diagram. |
| Table 62: Pattern Checker DRP Read-Only Registers | Updated description of CH[0/1]_RX_PRBSCNT. |
| Using RX Pattern Checker | Updated steps to enable and disable RX pattern checker, and added section for calculating bit error rate. |
| Table 74: GTM Transceiver PCB Design Checklist | Updated recommendations for MGTREFCLKP/MGTREFCLKN. |
| **07/26/2019 Version 1.1** | |
| Features | Updated list of protocols. |
| LCPLL | Updated FractionalDivider and FractionalPart calculations. |
| Dynamic Fractional (Frac-N) Divider | Added new topic. |
| PLL Power Down | Updated description of PLLPD signal. |
| Digital Monitor | Updated port and attribute names and tables. |
| Table 36: Supported PRBS Patterns | Updated descriptions of PRBS-13 and PRBS-31 patterns. |
| TX Gray Encoder and RX Gray Encoder | Updated descriptions of TX_GRAY_ENDIAN and RX_GRAY_ENDIAN attributes. |
| TX Pre-Coder and RX Pre-Coder | Updated descriptions of TX_PRECODE_ENDIAN and RX_PRECODE_ENDIAN attributes. |

| Section | Revision Summary |
|---|---|
| Table 45: TX Configurable Driver Ports | Updated descriptions of CH[0/1]_TXINHIBIT, CH[0/1]_TXEMPMAIN[5:0], CH[0/1]_TXEMPPRE[4:0], CH[0/1]_TXEMPPRE2[3:0], CH[0/1]_TXEMPPOST[4:0], and CH[0/1]_TXCTLFIRDAT[5:0]. |
| Table 51: RX Equalizer Attributes | Updated descriptions of CH[0/1]_RX_APT_CFG27A/B[15:0] and CH[0/1]_RX_APT_CFG28A/B[15:0]. |
| Table 64: RX Buffer Ports | Updated CH[0/1]_RXBUFSTATUS[2:0] port name and description. |
| Analog Power Supply Pins | Updated description of analog and digital power supplies. |
| PCB Design Checklist | Added note for VCCINT_GT -2LE devices. |
| **01/04/2019 Version 1.0** | |
| Initial release. | N/A |

# Table of Contents

XILINX

*Chapter 1*

# Transceiver and Tool Overview

## Introduction to the UltraScale Architecture

The Xilinx® UltraScale™ architecture is the first ASIC-class architecture to enable multi-hundred gigabit-per-second levels of system performance with smart processing, while efficiently routing and processing data on-chip. UltraScale architecture-based devices address a vast spectrum of high-bandwidth, high-utilization system requirements by using industry-leading technical innovations, including next-generation routing, ASIC-like clocking, 3D-on-3D ICs, multiprocessor SoC (MPSoC) technologies, and new power reduction features. The devices share many building blocks, providing scalability across process nodes and product families to leverage system-level investment across platforms.

Virtex® UltraScale+™ devices provide the highest performance and integration capabilities in a FinFET node, including both the highest serial I/O and signal processing bandwidth, as well as the highest on-chip memory density. As the industry's most capable FPGA family, the Virtex UltraScale+ devices are ideal for applications including 1+ Tb/s networking and data center and fully integrated radar/early-warning systems.

Virtex® UltraScale™ devices provide the greatest performance and integration at 20 nm, including serial I/O bandwidth and logic capacity. As the industry's only high-end FPGA at the 20 nm process node, this family is ideal for applications including 400G networking, large scale ASIC prototyping, and emulation.

Kintex® UltraScale+™ devices provide the best price/performance/watt balance in a FinFET node, delivering the most cost-effective solution for high-end capabilities, including transceiver and memory interface line rates as well as 100G connectivity cores. Our newest mid-range family is ideal for both packet processing and DSP-intensive functions and is well suited for applications including wireless MIMO technology, Nx100G networking, and data center.

Kintex® UltraScale™ devices provide the best price/performance/watt at 20 nm and include the highest signal processing bandwidth in a mid-range device, next-generation transceivers, and low-cost packaging for an optimum blend of capability and cost-effectiveness. The family is ideal for packet processing in 100G networking and data centers applications as well as DSP-intensive processing needed in next-generation medical imaging, 8k4k video, and heterogeneous wireless infrastructure.

Zynq® UltraScale+™ MPSoC devices provide 64-bit processor scalability while combining real-time control with soft and hard engines for graphics, video, waveform, and packet processing. Integrating an ARM®-based system for advanced analytics and on-chip programmable logic for task acceleration creates unlimited possibilities for applications including 5G Wireless, next generation ADAS, and Industrial Internet-of-Things.

This user guide describes the UltraScale architecture GTM transceivers and is part of the UltraScale architecture documentation suite available at: www.xilinx.com/ultrascale.

# Features

The GTM transceiver in the UltraScale+ FPGA is a high performance transceiver, supporting line rates between 10.3 Gb/s and 58 Gb/s. Based on the available PLL divider configurations in the GTM transceivers, the following line rates are supported:

- PAM4 modulation:
  - 58 Gb/s – 39.2 Gb/s
  - 29 Gb/s – 20.6 Gb/s
- NRZ modulation:
  - 29 Gb/s – 19.6 Gb/s
  - 14.5 Gb/s – 10.3 Gb/s

The GTM transceiver is Xilinx's first PAM4 enabled transceiver that is highly configurable and tightly integrated with the programmable logic resources of the FPGA. The table below summarizes the features by functional group that support a wide variety of applications.

*Table 1:* **GTM Transceiver Features**

| Group | Feature |
|---|---|
| PCS | KP4 Reed-Solomon forward error correction (RS-FEC) for up to 2 x 58 Gb/s or 1 x 116 electrical and optical links |
| | PRBS generator and checker |
| | Programmable FPGA logic interface |

*Table 1:* **GTM Transceiver Features** *(cont'd)*

| Group | Feature |
|---|---|
| PMA | LC tank oscillator PLL (LCPLL) for best jitter performance |
| | Flexible clocking with one PLL per Dual (two channels) |
| | Programmable TX output |
| | TX FIR filter with de-emphasis controls |
| | Continuous time linear equalizer (CTLE) |
| | Decision feedback equalization (DFE) |
| | Feed forward equalization (FFE) |

**Notes:**

1. A dual is a cluster or set of two GTM transceiver channels. One GTM_DUAL primitive, one differential reference clock pin pair, and analog supply pins. There is no channel primitive.

The GTM transceiver supports NRZ and PAM4 modulation as well as the following protocols:

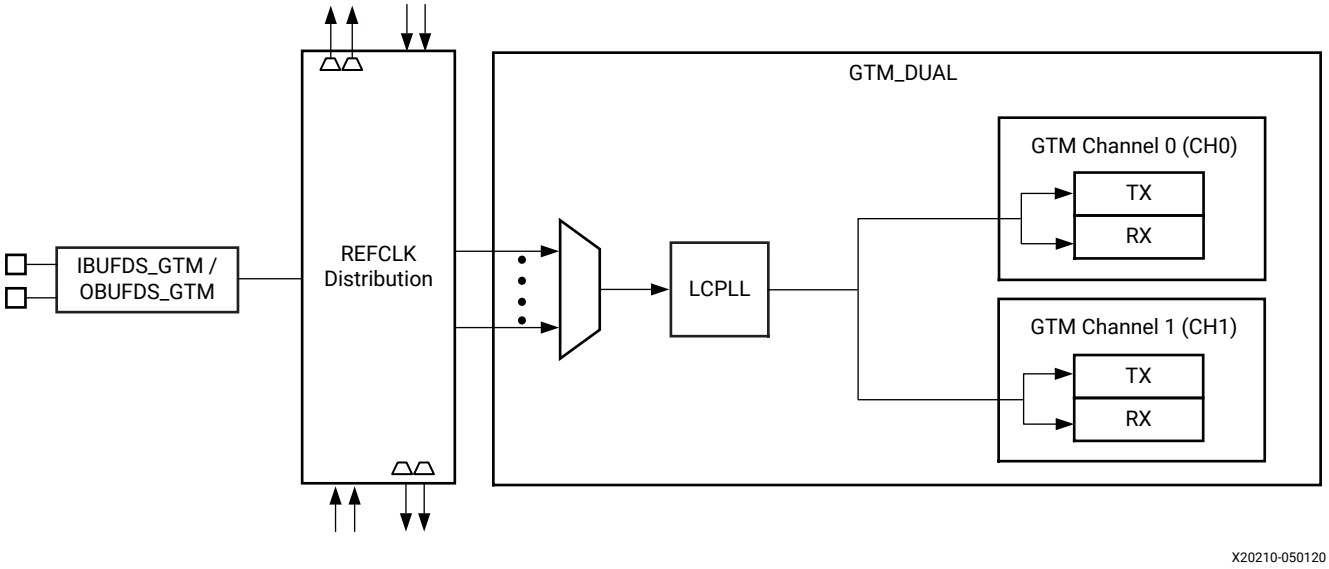
- 400GAUI-8
- 200GAUI-4
- 100GAUI-2
- 50GAUI-1
- 200GBASE-KR2
- 200GBASE-CR2
- 100GBASE-KR2
- 100GBASE-CR2
- 50GBASE-KR
- 50GBASE-CR
- CAUI-4
- 50GAUI-2
- LAUI-2
- 25GAUI-1
- 100GBASE-KR4
- 100GBASE-CR4
- 25GBASE-KR
- 25GBASE-CR
- XLPPI
- 40GBASE-CR4

- 10GBASE-CR4
- 10GBASE-SR/LR
- 10GBASE-KR
- CEI-56G-VSR/MR/LR-PAM4
- CEI-25G/28G-VSR/MR/LR
- CEI-11G-SR/MR/LR
- 10G-15G backplane capability
- Ethernet AN/LT (autonegotiation/link training)
- 28.21G PAM4 backplanes
- 58G PAM4 backplanes
- OTU4 optical modules NRZ optical7% FEC
- OTU4 optical module PAM4 optical7% FEC
- Interlaken 25.78125G
- Interlaken 12.5G
- CPRI 48G PAM4

***Note*:** See the UltraScale device data sheets (see http://www.xilinx.com/documentation) for a complete list of electrical compliant protocols.

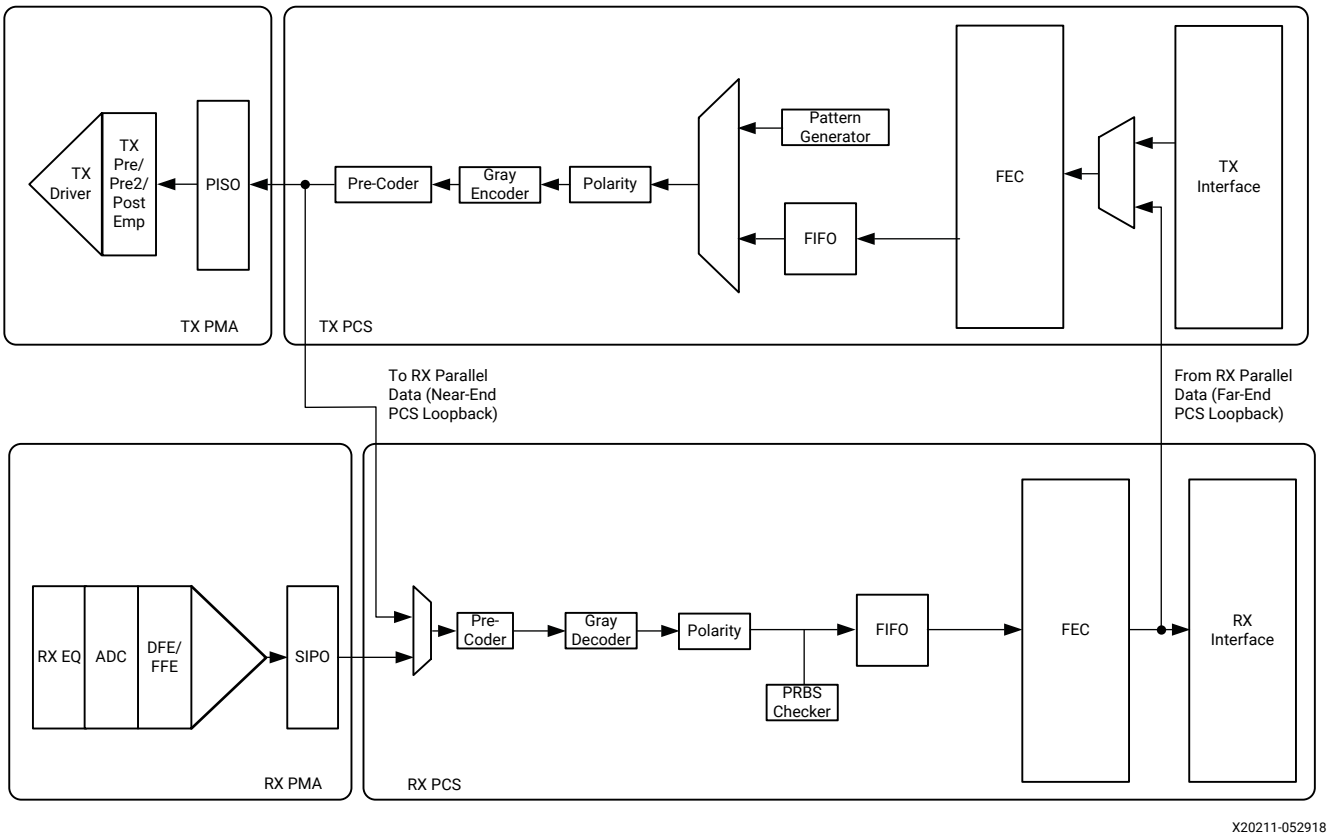
The first-time user is recommended to read *High-Speed Serial I/O Made Simple*, which discusses high-speed serial transceiver technology and its applications. The Xilinx Vivado® IP catalog includes an UltraScale+ FPGAs GTM Transceivers Wizard to automatically configure GTM transceivers to support configurations for different protocols and perform custom configurations. The GTM transceiver offers a data rate range and features that allow physical layer support for various protocols. The following figure illustrates the clustering of one GTM_DUAL primitive.

*Figure 1:* **GTM Dual Configuration**




The GTM_DUAL primitive contains one LCPLL and two GTM channels. Contrary to other UltraScale+ device transceivers such as the GTH and GTY transceivers, the GTM transceiver does not contain channel/common primitives. All channel ports and attributes are within the GTM_DUAL primitive. The following figure illustrates the topology of a GTM channel.

*Figure 2:* **GTM Channel Topology**




# UltraScale+ FPGAs GTM Transceivers Wizard

The UltraScale+ FPGAs GTM Transceivers Wizard (hereinafter called the Wizard) is the preferred tool to generate a wrapper to instantiate the GTM_DUAL. The Wizard is located in the IP catalog under the IO Interfaces category.

**RECOMMENDED:** *Download the most recent IP update before using the Wizard. Details on how to use this Wizard can be found in Virtex UltraScale+ FPGAs GTM Transceivers Wizard LogiCORE IP Product Guide (PG315).*

# Simulation

The simulation environment and the test bench must fulfill specific prerequisites before running simulation using the GTM_DUAL primitives. For instructions on how to set up the simulation environment for supported simulators depending on the used hardware description language (HDL), see the latest version of the *Virtex UltraScale+ FPGAs GTM Transceivers Wizard LogiCORE IP Product Guide* (PG315) and *Vivado Design Suite User Guide: Logic Simulation* (UG900).

The prerequisites for simulating a design with the GTM_DUAL primitives are listed:

- A simulator with support for SecureIP models: SecureIP is an IP encryption methodology. SecureIP models are encrypted versions of the System Verilog HDL used for implementation of the modeled block. To support SecureIP models, a simulator that complies with the encryption standards described in the Verilog language reference manual (LRM)—IEEE Standard for Verilog Hardware Description Language (IEEE Std 1364-2005) is required.
- A mixed-language simulator for VHDL simulation: SecureIP models use a System Verilog standard. To use them in a VHDL design, a mixed-language simulator is required. The simulator must be able to simulate VHDL and Verilog simultaneously.
- An installed GTM transceiver SecureIP model.
- The correct setup of the simulator for SecureIP use (initialization file, environment variables).
- The correct simulator resolution (Verilog).

## Ports and Attributes

There are no simulation-only ports on the GTM_DUAL primitives. The GTM_DUAL primitive has attributes intended only for simulation. The following table lists the simulation-only attributes of the GTM_DUAL primitive. The names of these attributes start with SIM_.

*Table 2:* **GTM_DUAL Simulation-Only Attributes**

| Attribute | Type | Description |
|---|---|---|
| SIM_RESET_SPEEDUP | String | If the SIM_RESET_SPEEDUP attribute is set to TRUE (default), an approximated reset sequence is used to speed up the reset time for simulations, where faster reset times and faster simulation times are desirable. If the SIM_RESET_SPEEDUP attribute is set to FALSE, the model emulates hardware reset behavior in detail. |
| SIM_DEVICE | String | This attribute selects the simulation version to match different versions of silicon. The default for this attribute is ULTRASCALE_PLUS. |

# Implementation

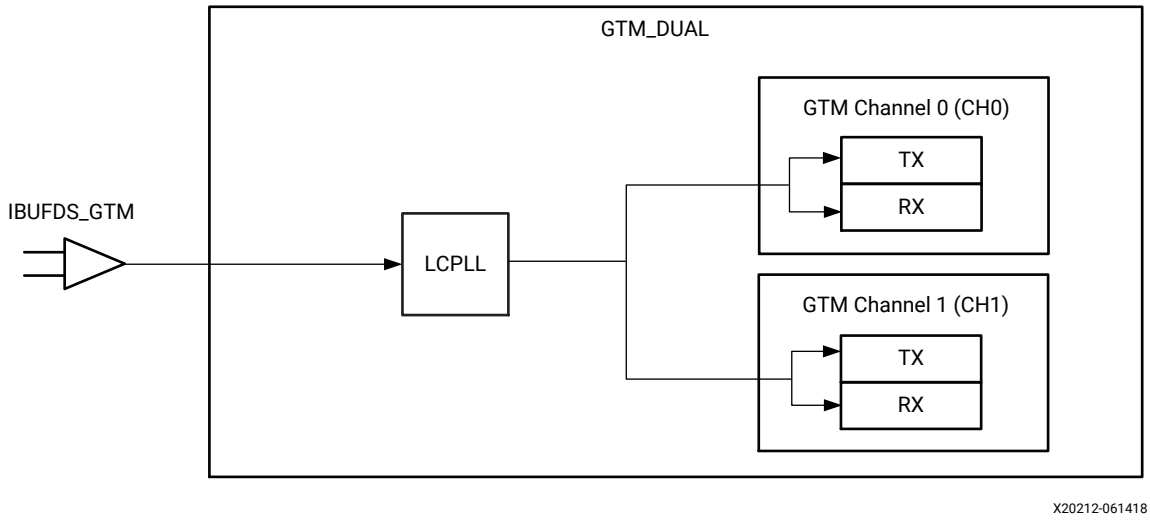
It is a common practice to define the location of GTM transceiver Duals early in the design process to ensure correct usage of clock resources and to facilitate signal integrity analysis during board design. The implementation flow facilitates this practice through the use of location constraints in the XDC file.

The position of each GTM transceiver Dual primitive is specified by an XY coordinate system that describes the column number and the relative position within that column. For a given device/package combination, the transceiver with the coordinates X0Y0 is located at the lowest position of the lowest available bank.

There are two ways to create an XDC file for designs that utilize the GTM transceivers. The preferred method is to use the UltraScale+ FPGAs GTM Transceivers Wizard. The Wizard automatically generates XDC file templates that configure the transceivers and contain placeholders for GTM transceiver placement information. The XDC files generated by the Wizard can then be edited to customize operating parameters and placement information for the application.

The second approach is to create the XDC file manually. When using this approach, you must enter both configuration attributes that control transceiver operation as well as the location parameters. Care must be taken to ensure that all of the parameters needed to configure the GTM transceiver are correctly entered. A GTM_DUAL primitive must be instantiated as shown in the following figure.

*Figure 3:* **One-Dual, Two-Channel Configuration (Reference Clock from the LCPLL)**




Each dual contains an LCPLL. Therefore, a reference clock can be connected directly to a GTM_DUAL primitive.

# Shared Features

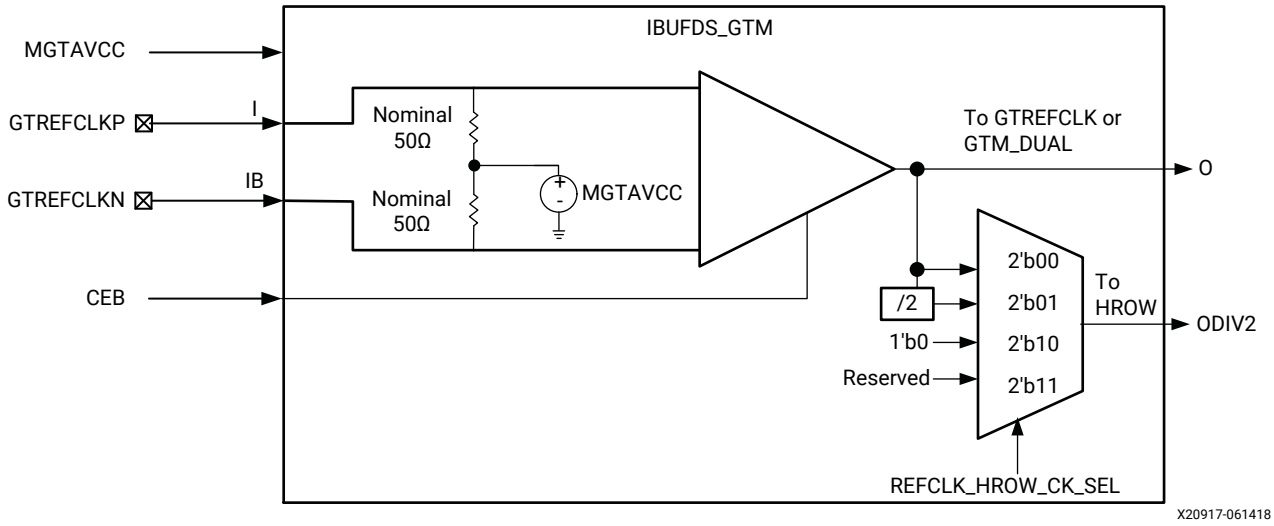
## Reference Clock Input/Output Structure

The reference clock structure in the GTM transceiver supports two modes of operation: input mode and output mode. In the input mode of operation, your design provides a clock on the dedicated reference clock I/O pins that are used to drive the LCPLL. In the output mode of operation, the recovered clocks (RXRECCLK0 and RXRECCLK1) from any of the two channels within the same Dual can be routed to the dedicated reference clock I/O pins. This output clock can then be used as the reference clock input at a different location. The mode of operation cannot be changed during run time.

### Input Mode

The reference clock input mode structure is illustrated in the following figure. The input is terminated internally with 50Ω on each leg to MGTAVCC. The reference clock is instantiated in software with the IBUFDS_GTM software primitive. The ports and attributes controlling the reference clock input are tied to the IBUFDS_GTM software primitive.

*Figure 4:* **Reference Clock Input Structure**



### *Ports and Attributes*

The following table defines the reference clock input ports in the IBUFDS_GTM software primitive.

*Table 3:* **Reference Clock Input Ports (IBUFDS_GTM)**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CEB | In | N/A | This is the active-Low asynchronous clock enable signal for the clock buffer. Setting this signal High powers down the clock buffer. |
| I | In (pad) | N/A | These are the reference clock input ports that get mapped to GTREFCLKP. |
| IB | In (pad) | N/A | These are the reference clock input ports that get mapped to GTREFCLKN. |
| O | Out | N/A | This output drives the GTREFCLK signal in the GTM_DUAL software primitive. Refer to Reference Clock Selection and Distribution for more details. |
| ODIV2 | Out | N/A | This output can be configured to output either the O signal or a divide-by-2 version of the O signal. It can drive the BUFG_GT via the HROW routing. Refer to Reference Clock Selection and Distribution for more details. |

The following table defines the attributes in the IBUFDS_GTM software primitive that configure the reference clock input.

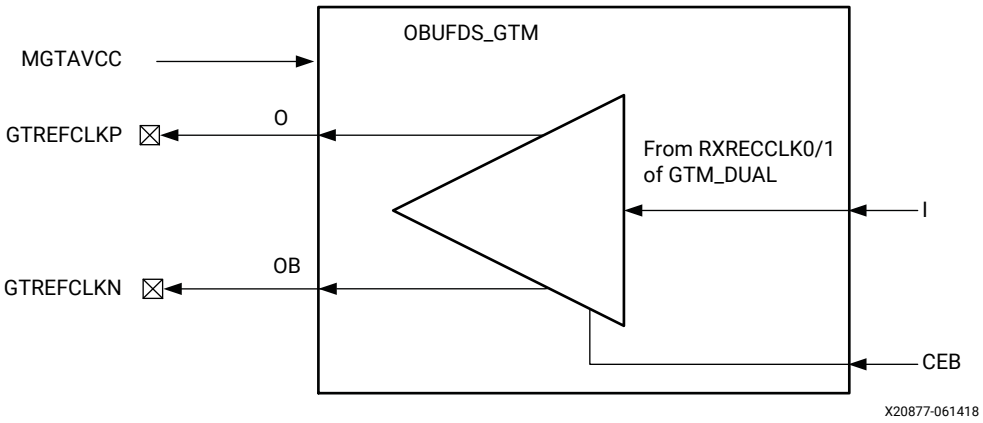
*Table 4:* **Reference Clock Input Attributes (IBUFDS_GTM)**

| Attribute | Type | Description |
|---|---|---|
| REFCLK_EN_TX_PATH | 1-bit | Reserved. This attribute must always be set to `1'b0`. |
| REFCLK_HROW_CK_SEL | 2-bit | Configures the ODIV2 output port:<br>`2'b00`: ODIV2 = O.<br>`2'b01`: ODIV2 = Divide-by-2 version of O.<br>`2'b10`: ODIV2 = `1'b0`.<br>`2'b11`: ODIV2 = Reserved. |
| REFCLK_ICNTL_RX | 2-bit | Reserved. Use the recommended value from the Wizard. |

# Output Mode

The reference clock output mode can be accessed via the OBUFDS_GTM software primitive. The reference clock output mode structure for the OBUFDS_GTM primitive is shown in the following figure. The ports and attributes controlling the reference clock output are tied to the OBUFDS_GTM software primitive.

*Figure 5:* **Reference Clock Output Use Model for OBUFDS_GTM**




### Ports and Attributes

The following table defines the ports in the OBUFDS_GTM software primitive.

*Table 5:* **Reference Clock Output Ports (OBUFDS_GTM)**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CEB | In | N/A | This is the active-Low asynchronous clock enable signal for the clock buffer. Setting this signal High powers down the clock buffer. |
| I | In (pad) | N/A | Recovered clock input. Connect to the output port RXRECCLK0/1 of the GTM_DUAL primitive. |
| O | In (pad) | N/A | Reference clock output port that gets mapped to GTREFCLKP. |
| OB | Out | N/A | Reference clock output port that gets mapped to GTREFCLKN. |

The following table defines the attributes in the OBUFDS_GTM software primitive that configure the reference clock output.

*Table 6:* **Reference Clock Output Attributes (OBUFDS_GTM)**

| Attribute | Type | Description |
|---|---|---|
| REFCLK_EN_TX_PATH | 1-bit | Reserved. This attribute must always be set to `1'b1`. |
| REFCLK_ICNTL_TX | 5-bit | Reserved. Use the recommended value from the Wizard. |

# Reference Clock Selection and Distribution

The GTM transceivers in Virtex UltraScale+ FPGAs provide different reference clock input options. Clock selection and availability is similar to the GTY transceivers in UltraScale+ devices, but the reference clock selection architecture supports only one LCPLL shared per Dual (two GTM transceiver channels).

From an architecture perspective, a Dual contains a grouping of two GTM channels inside one GTM_DUAL primitive, one dedicated external reference clock pin pair, and dedicated reference clock routing. The reference clock for a GTM_DUAL primitive must also be instantiated. For duals operating at line rates lower than 14.5 Gb/s (NRZ) and 29 Gb/s (PAM4), the reference clock for a Dual can also be sourced from the Dual above via GTNORTHREFCLK or from the Dual below via GTSOUTHREFCLK. For devices that support stacked silicon interconnect (SSI) technology, the reference clock sharing via the GTNORTHREFCLK and GTSOUTHREFCLK ports is limited within its own super logic region (SLR). Duals operating at line rates above 14.5 Gb/s (NRZ) and 29 Gb/s (PAM4) should not source a reference clock from another Dual.

See the UltraScale device data sheets (see http://www.xilinx.com/documentation) for more information about SSI technology.

Reference clock features include:

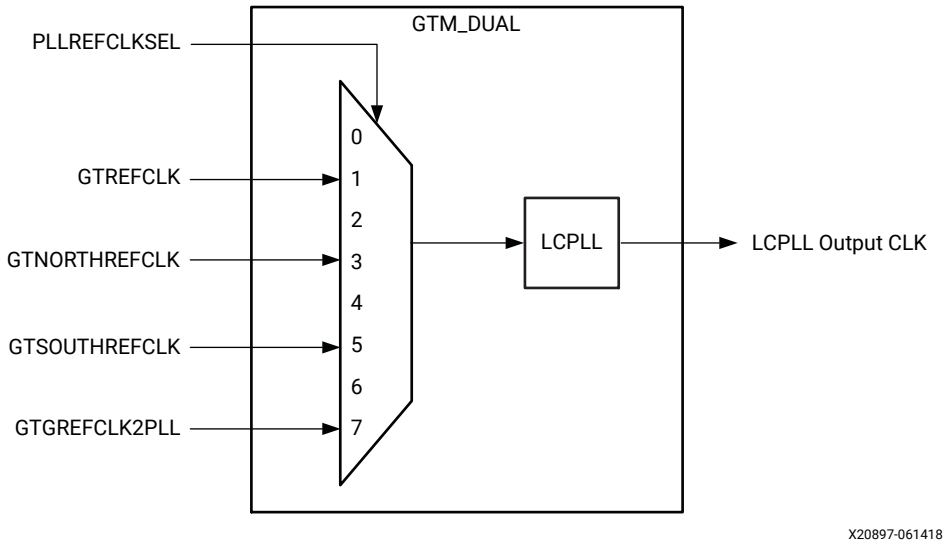
- Clock routing for northbound and southbound clocks.
- Flexible clock inputs available for the LCPLL.
- Static or dynamic selection of the reference clock for the LCPLL.

The Dual architecture has two GTM transceivers, one dedicated reference clock pin pair, and dedicated north and south reference clock routing. Each GTM dual has three clock pair inputs available:

- One local reference clock pin pair, GTREFCLK.
- One reference clock pin pair for the Dual above, GTSOUTHREFCLK.
- One reference clock pin pair from the Dual below, GTNORTHREFCLK.

The figure below shows the detailed view of a reference clock multiplexer structure within a single GTM_DUAL primitive. The PLLREFCLKSEL port is required when multiple reference clock sources are connected to this multiplexer. A single reference clock is most commonly used. In the case of a single reference clock, connect the reference clock to the GTREFCLK ports and tie the PLLREFCLKSEL ports to `3'b001`. The Xilinx software tools handle the complexity of the multiplexers and associated routing.

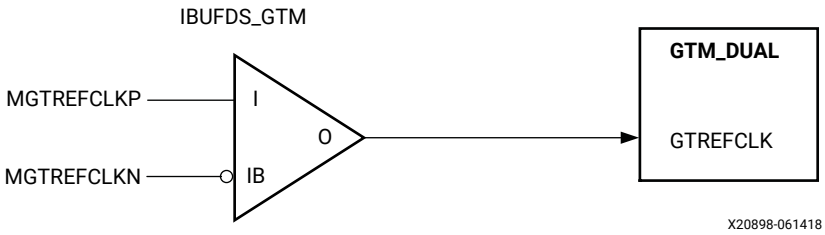
*Figure 6:* **LCPLL Reference Clock Selection Multiplexer**




# Single External Reference Clock Use Model

Each Dual has one set of dedicated differential reference clock input pins (MGTREFCLK[P/N]) that can be connected to the external clock sources. In a single external reference clock use model, an IBUFDS_GTM must be instantiated to use the dedicated differential reference clock source. The following figure shows a single external reference clock connected to the LCPLL inside the Dual. The user design connects the IBUFDS_GTM output (O) to the GTREFCLK ports of GTM_DUAL.

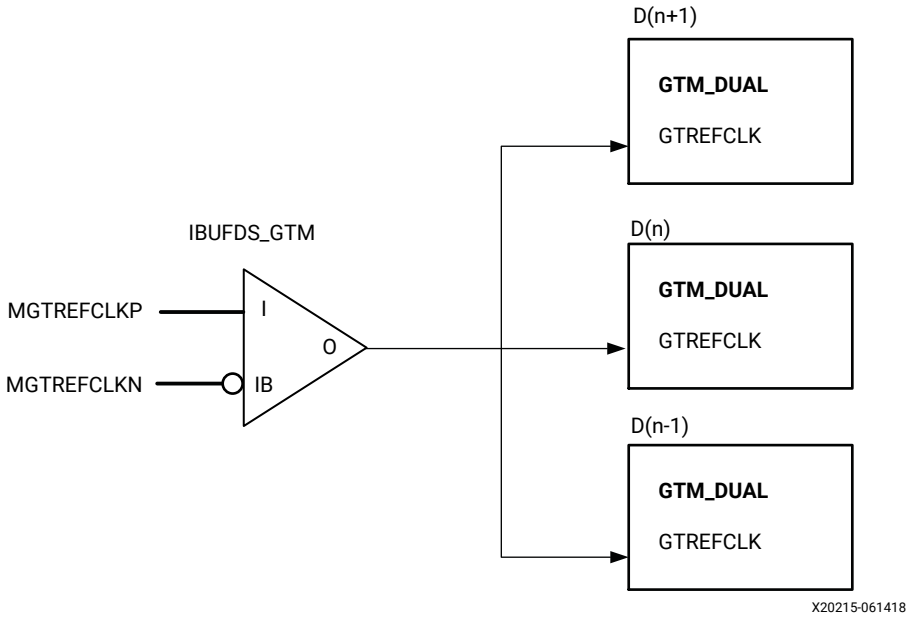
*Figure 7:* **Single External Reference Clock in a Dual**




***Note:*** The IBUFDS_GTM diagram in the above figure is a simplification. The output port ODIV2 is left floating, and the input port CEB is set to logic 0.

The following figure shows a single external reference clock with multiple Duals connected. The user design connects the IBUFDS_GTM output (O) to the GTREFCLK ports of the GTM_DUAL primitives. In this case, the Xilinx implementation tools make the necessary adjustments to the north/south routing as well as the pin swapping necessary to route the reference clock from one Dual to another when required.

*Figure 8:* **Single External Reference Clock with Multiple Duals**




***Note:*** The IBUFDS_GTM diagram in the above figure is a simplification. The output port ODIV2 is left floating, and the input port CEB is set to logic 0.

These rules must be observed when sharing a reference clock to ensure that jitter margins for high-speed designs are met:

- The number of Duals above the sourcing Dual must not exceed one.
- The number of Duals below the sourcing Dual must not exceed one.
- The total number of Duals sourced by an external clock pin pair (MGTREFCLKP/MGTREFCLKN) must not exceed three Duals.

The maximum number of Duals that can be sourced by a single clock pin pair is three (six transceivers). Designs with more than three Duals require the use of multiple external clock pins to ensure that the rules for controlling jitter are followed. When multiple clock pins are used, an external buffer can be used to drive them from the same oscillator.

**IMPORTANT!** *Upon device configuration, the clock output from the IBUFDS_GTM which takes inputs from MGTREFCLKP and MGTREFCLKN can only be used as long as the GTPOWERGOOD signal has already asserted High.*

## Ports and Attributes

The following table defines the clocking ports and attributes for the GTM_DUAL primitive.

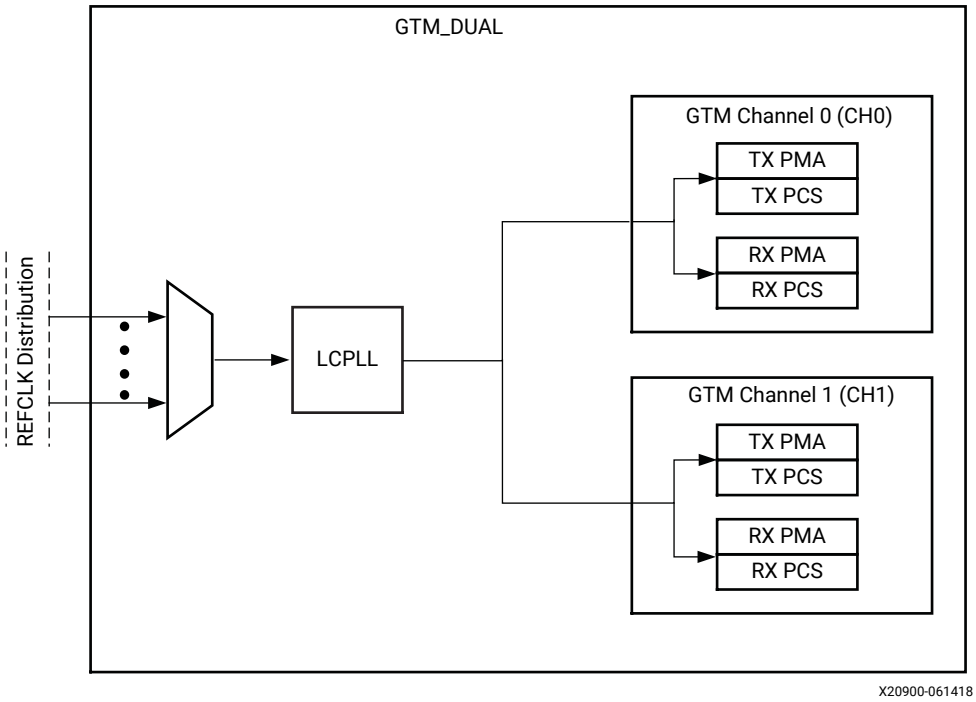
*Table 7:* **GTM_DUAL Clocking Ports**

| Port | Direction | Clock Domain | Description |
|---|---|---|---|
| PLLREFCLKSEL[2:0] | In | Async | Input to dynamically select the input reference clock to the LCPLL. Set this input to `3'b001` and connect to GTREFCLK when only one clock source is connected to the PLL reference clock selection multiplexer:<br>`3'b000`: Reserved.<br>`3'b001`: GTREFCLK selected.<br>`3'b010`: Reserved.<br>`3'b011`: GTNORTHREFCLK selected.<br>`3'b100`: Reserved.<br>`3'b101`: GTSOUTHREFCLK.<br>`3'b110`: Reserved.<br>`3'b111`: GTGREFCLK2PLL. |
| GTGREFCLK2PLL | In | Clock | Reference clock generated by the internal interconnect logic. This input is reserved for internal testing purposes only. |
| GTREFCLK | In | Clock | External clock driven by IBUFDS_GTM for the LCPLL. |
| GTNORTHREFCLK | In | Clock | Northbound clock from the Dual below. |
| GTSOUTHREFCLK | In | Clock | Southbound clock from the Dual above. |
| PLLREFCLKLOST | Out | Async | A High on this signal indicates that the reference clock to the phase frequency detector of the LCPLL is lost. |
| PLLREFCLKMONITOR | Out | Clock | LCPLL reference clock selection multiplexer output. |

# LCPLL

Each Dual contains one LC-based PLL, referred to as LCPLL, and cannot be shared with neighboring Duals. The internal clocking architecture of the GTM Dual is shown in the following figure.

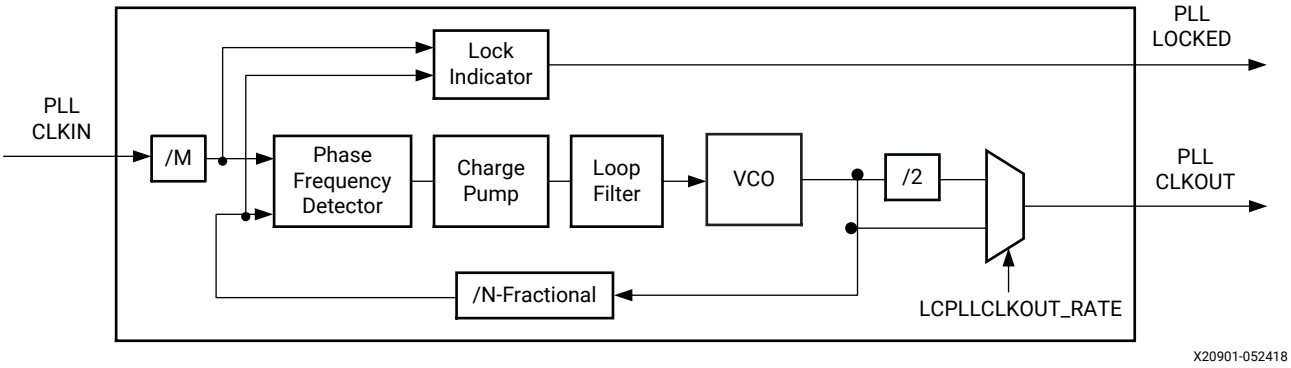
*Figure 9:* **Internal Dual Clocking Architecture**




The LCPLL input clock selection is described in Reference Clock Selection and Distribution. The LCPLL outputs feed the TX and RX clock divider blocks, which control the generation of serial and parallel clocks used by the PMA and PCS blocks. The LCPLL is shared between the TX and RX datapaths.

The figure below illustrates a conceptual view of the LCPLL architecture. The input clock can be divided by a factor of M before it is fed into the phase frequency detector. The feedback divider N determines the voltage-controlled oscillator (VCO) multiplication ratio. For line rates below 28.21 Gb/s (NRZ) and 56.42 Gb/s (PAM4), a fractional-N divider is supported where the effective ratio is a combination of the N factor plus a fractional part. The LCPLL output frequency depends on the settings of LCPLLCLKOUT_RATE. When LCPLLCLKOUT_RATE is set to HALF, the output frequency is half of the VCO frequency. When it is set to FULL, the output frequency is the same as the VCO frequency. A lock indicator block compares the frequencies of the reference clock and VCO feedback clock to determine if a frequency lock has been achieved.

*Figure 10:* **Internal Dual Clocking Architecture**




The LCPLL VCO operates within 9.8 GHz–14.5 GHz. The Xilinx software tool chooses the appropriate LCPLL setting based on application requirements. The following equation shows how to determine the PLL output frequency (GHz).

*Equation 1:* **$f_{PLLClkout}$ Calculation**

$$f_{PLLClkout} = f_{PLLClkin} * \frac{N + FractionalDivider}{M * LCPLLCLKOUT\_RATE}$$

The following equation shows how to determine the line rate (Gb/s).

*Equation 2:* **$f_{LineRate}$ Calculation**

$$f_{LineRate} = f_{PLLClkout} * Modulation$$

The following equations show how to determine the fractional divider presented in Equation 1: fPLLClkout Calculation.

*Equation 3:* **FractionalDivider Calculation**

$$FractionalDivider = N_{SDM} + FractionalPart$$

*Equation 4:* **FractionalPart Calculation**

$$FractionalPart = \frac{SDMDATA}{2^{SDMWIDTH}}$$

The table below lists the allowable values.

*Table 8:* **LCPLL Divider Settings**

| Factor | Attribute or Port | Valid Settings |
|---|---|---|
| M | LCPLL_REFCLK_DIV | 1, 2, 3, 4 |
| N | PLLFBDIV[7:0] | 16–160 (Integer only) |

*Table 8:* **LCPLL Divider Settings** *(cont'd)*

| Factor | Attribute or Port | Valid Settings |
|---|---|---|
| LCPLLCLKOUT_RATE | LCPLLCLKOUT_RATE | `1'b1`: 1 (Full), `1'b0`: 2 (Half) |
| Modulation | See TX Configurable Driver | 2 (NRZ), 4 (PAM4) |
| $N_{SDM}$ | SDMDATA[SDMWIDTH + 1:SDMWIDTH] | Integer part of fractional divider. (Twos complement) –2, –1, 0, 1 |
| SDMDATA | SDMDATA[SDMWIDTH – 1:0] | Fractional part of fractional divider. 0 – ($2^{24}$ – 1) |
| SDMWIDTH | SDM_WIDTHSEL | 16, 20, 24 |

# Ports and Attributes

The following tables define the LCPLL ports and attributes, respectively.

*Table 9:* **LCPLL Ports**

| Port | Direction | Clock Domain | Description |
|---|---|---|---|
| GTGREFCLK2PLL | In | Clock | Reference clock generated by the internal interconnect logic. This input is reserved for internal testing purposes. |
| PLLFBCLKLOST | Out | PLLMONCLK | A High on this signal indicates the feedback clock from the LCPLL feedback divider to the phase frequency detector of the LCPLL is lost |
| PLLFBDIV[7:0] | In | Async | PLL feedback divider selection. Actual feedback divider value is PLLFBDIV + 2. Valid values are from 14–158. (Actual divider values are 16–160.) |
| PLLLOCK | Out | Async | This active-High LCPLL frequency lock signal indicates that the LCPLL frequency is within a predetermined tolerance. The transceiver and its clock outputs are not reliable until this condition is met. |
| PLLMONCLK | In | Clock | Stable reference clock for the detection of the feedback and reference clock signals to the LCPLL. The input reference clock to the LCPLL or any output clock generated from the LCPLL must not be used to drive this clock. This clock is required only when using the PLLFBCLKLOST and PLLREFCLKLOST ports. It does not affect the LCPLL lock detection, reset, and power-down functions. |
| PLLPD | In | Async | An active-High signal powers down the LCPLL. |
| PLLREFCLKLOST | Out | PLLMONCLK | A High on this signal indicates the reference clock to the phase frequency detector of the LCPLL is lost. |
| PLLREFCLKMONITOR | Out | Clock | PLL reference clock selection multiplexer output. |
| PLLRESET | In | Async | This port is driven High and then deasserted to start the LCPLL reset. |
| PLLRESETBYPASSMODE | In | Async | Reserved. Tied Low. |
| PLLRESETDONE | Out | cfg_mclk | Status signal that indicates when the PLL reset sequence is complete. |
| PLLRESETMASK[1:0] | In | Async | Bit 0 enables bit mask for PLL reset. Bit 1 enables bit mask for PLL SDM reset. |
| PLLRSVDIN[15:0] | In | Reserved | Reserved. This port must be set to `0x0000`. |

*Table 9:* **LCPLL Ports** *(cont'd)*

| Port | Direction | Clock Domain | Description |
|---|---|---|---|
| PLLRSVDOUT | Out | Async | Reserved. |
| BGBYPASSB | In | Async | Reserved. This port must be set to `1'b1`. Do not modify this value. |
| BGMONITORENB | In | Async | Reserved. This port must be set to `1'b1`. Do not modify this value. |
| BGPDB | In | Async | Reserved. This port must be set to `1'b1`. Do not modify this value. |
| BGRCALOVRD[4:0] | In | Async | Reserved. This port must be set to `5'b00000`. Do not modify this value. |
| BGRCALOVRDENB | In | Async | Reserved. This port must be set to `1'b1`. Do not modify this value. |
| RCALEN | In | Async | Reserved. This port must be set to `1'b1`. Do not modify this value. |
| SDMDATA[25:0] | In | Async | Input to set the fractional divider. Bits [SDMWIDTH + 1:SDMWIDTH] are the integer part of the divider in twos complement. Bits [SDMWIDTH – 1:SDMWIDTH] set the fractional part of the divider. |
| SDMTOGGLE | In | Async | Reserved. Set to `1'b0`. |

*Table 10:* **LCPLL Attributes**

| Attributes | Type | Description |
|---|---|---|
| BIAS_CFG0 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| BIAS_CFG1 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| BIAS_CFG2 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| BIAS_CFG3 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| BIAS_CFG4 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| A_CFG | 16-bit | Reserved. Use the recommended value from the Wizard. |
| CRS_CTRL_CFG0 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| CRS_CTRL_CFG1 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| DRPEN_CFG | 16-bit | Reserved. Use the recommended value from the Wizard. |
| PLL_CFG0 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Address** | **Description** |
| LCPLLCLKOUT_RATE | [8] | Sets the LCPLLCLKOUT_RATE factor either to provide full LCPLL VCO frequency, or half of LCPLL VCO frequency at the output:<br>`1'b0`: Half rate.<br>`1'b1`: Full rate. |
| PLL_CFG1 | 16-bit | Reserved. Use the recommended value from the Wizard. |

*Table 10:* **LCPLL Attributes** *(cont'd)*

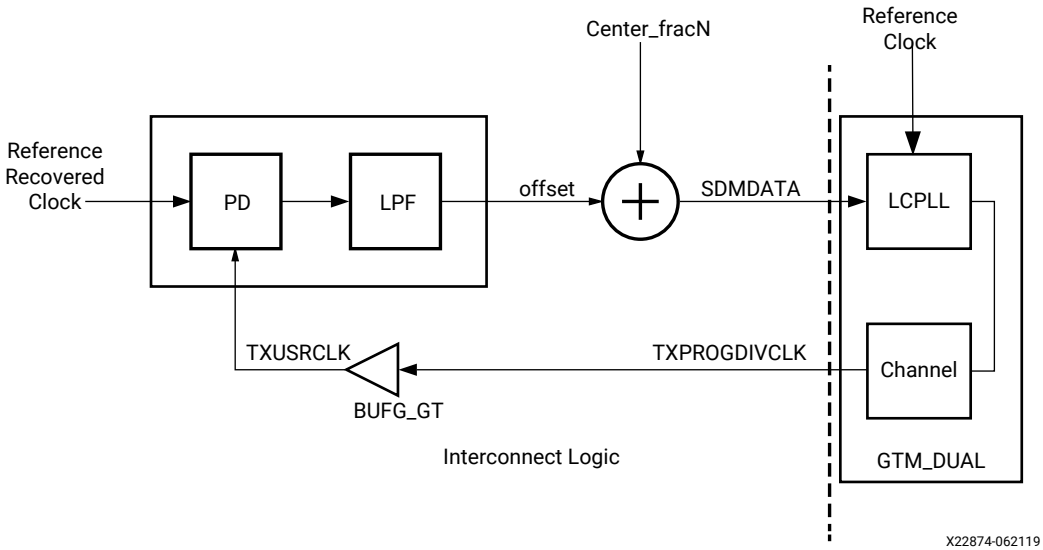
| Attributes | Type | Description |
|---|---|---|
| **Bit Name** | **Address** | **Description** |
| LCPLL_REFCLK_DIV | [4:0] | LCPLL reference clock divider M settings.<br>`5'b00000`: Div by 2.<br>`5'b00001`: Div by 3.<br>`5'b00010`: Div by 4.<br>`5'b10010`: Div by 1.<br>Other values are not valid. |
| PLL_CFG2 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| PLL_CFG3 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| PLL_CFG4 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| PLL_CFG5 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| PLL_CFG6 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| SDM_CFG0[15:0] | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Address** | **Description** |
| SDM_BYPASS | [7] | This attribute enables the use of the SDM block (fractional divider). Set to `1'b0` to enable the SDM block. |
| SDM_WIDTHSEL | [10:9] | This attribute sets the denominator of the fractional part of the feedback divider:<br>`2'b00`: 24 bits.<br>`2'b01`: 20 bits.<br>`2'b10`: 16 bits.<br>`2'b11`: Reserved. |
| SDM_CFG1 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| SDM_CFG2 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| SDM_SEED_CFG0 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| SDM_SEED_CFG1 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| A_SDM_DATA_CFG0 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| A_SDM_DATA_CFG1 | 16-bit | Reserved. Use the recommended value from the Wizard. |

# Use Modes

## *Dynamic Fractional (Frac-N) Divider*

Typically, SDMDATA is statically set to create a static Frac-N divider. However, it is also possible to continually update the value of SDMDATA. The following figure shows how to use the GTM transceiver in UltraScale+ FPGAs when Dynamic Frac-N is required.

*Figure 11:* **Dynamic Frac-N Example**




The operation of the Dynamic Frac-N divider has a user operated strobe (SDMTOGGLE) pulse that controls the SDMDATA transfer from the fabric to the transceiver. The assumption here is the logic that drives SDMTOGGLE and SDMDATA is clocked by SYSTEM clock.

The following figures show the same operation with different relationships between LCPLL FBCLK and SYSTEM clock frequencies.

*Figure 12:* **FBCLK is Faster than SYSTEM Clock**

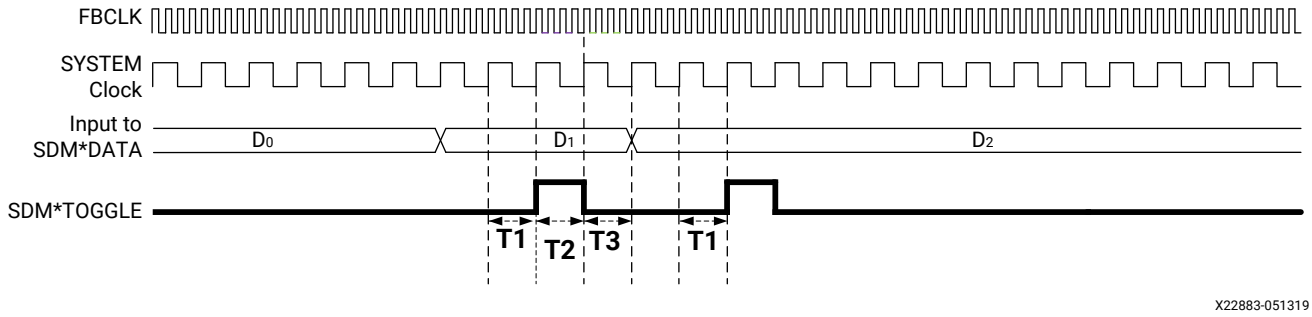
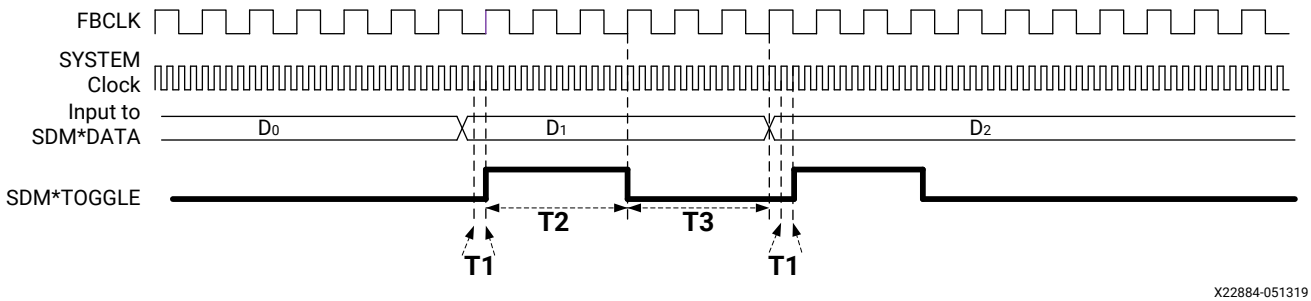



*Figure 13:* **FBCLK is Slower than SYSTEM Clock**



***Note*:** USRCLK or DRPCLK can be used as the SYSTEM clock.

There are three major timing requirements marked as T1, T2, and T3 in Figure 12 and Figure 13. These timing requirements are described in the following table.

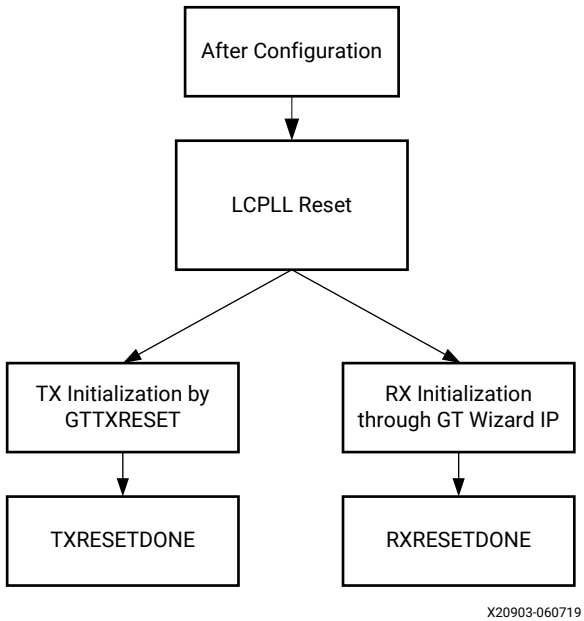
| Time Period | Requirements | Comments |
|---|---|---|
| T1 | ≥ One system clock cycle | This is the least amount of time SDMDATA must be *stable and valid before* SDMTOGGLE can be asserted. Period T1 should at least be one SYSTEM Clock cycle. During this period, SDMTOGGLE must be Low. |
| T2 | ≥ Three FBCLK cycles | This is the least amount of time SDMTOGGLE must be held High. Period T2 should be at least three FBCLK cycles. SDMDATA must not change during T2. |
| T3 | ≥ Three FBCLK cycles | This is the least amount of time SDMTOGGLE must be held Low. Period T3 should be at least three FBCLK cycles. SDMDATA must not change during T3. T3 does not include T1. |

***Note*:** The user must keep track of the frequency deviation with respect to the initial LCPLL VCO frequency before the dynamic update is initiated. If the frequency deviation is more than +/-200 ppm compared to the original VCO frequency, an LCPLL reset must be asserted after the dynamic update..

# Reset and Initialization

The GTM transceiver must be initialized after device power-up and configuration before it can be used. The GTM transmitter (TX) and receiver (RX) can be initialized independently and in parallel as shown in the following figure.

*Figure 14:* **GTM Transceiver Initialization Overview**



The GTM transceiver TX and RX initialization comprises three steps:

1. Initializing the associated LCPLL driving TX/RX
2. Initializing the TX/RX PMA datapath
3. Initializing the TX/RX PCS datapath

***Note*:** The RX PMA datapath reset must be executed through the GT Wizard IP. See *Virtex UltraScale+ FPGAs GTM Transceivers Wizard LogiCORE IP Product Guide* (PG315).

The TX and RX in the GTM transceiver receive the clock through the LCPLL in the transceiver's own Dual. In the power-on initialization sequence, the LCPLL used by the TX and RX must be initialized first. The LCPLL used by the TX and RX is reset individually and its reset operation independent from TX and RX resets. The TX and RX datapaths must be initialized only after the associated LCPLL is locked.

The GTM transceiver TX and RX use a state machine to control the initialization process. They are partitioned into a few reset regions. The partition allows the reset state machine to control the reset process in a sequence that the PMA can be reset first and the PCS can be reset after the assertion of the TXUSERRDY or RXUSERRDY. It also allows the PMA, the PCS, and the functional blocks inside them to be reset individually when needed during normal operation.

The GTM transceiver offers two types of reset: initialization and component.

- **Initialization Reset:** This reset is used to complete GTM transceiver initialization and must be used after device power-up and configuration. For the GTM TX, the initialization reset port is GTTXRESET. During normal operation the user can use this port for TX initialization reset. During TX initialization reset, TXRESETMODE should be set to sequential mode and all TX PMA and PCS bits of TXPMARESETMASK and TXPCSRESETMASK should be set to High. For the GTM RX, initialization reset must be executed using the GT Wizard IP. See *Virtex UltraScale+ FPGAs GTM Transceivers Wizard LogiCORE IP Product Guide* (PG315).
- **Component Reset:** This reset is used for special cases and specific subsection resets where the GTM transceiver is in normal operation. For the GTM TX, a component that is required to be reset is selected by the associated bit with TXPMARESETMASK and TXPCSRESETMASK. The TX component reset is triggered by toggling the GTTXRESET port. Separate asynchronous TX reset ports are also available for each component: TXCKALRESET, TXFECRESET, TXPCSRESET, and TXPMARESET. For the GTM RX, PMA component reset is not supported. For RX PCS, a component that is required to be reset is selected by the associated bit with RXPCSRESETMASK. An RX PCS component reset is triggered by toggling the GTRXRESET port. For proper RX PCS component reset, the user must set RXPMARESETMASK[7:0] to `8'b00000000` before toggling GTRXRESET. After RX PCS component reset, the user must set RXPMARESETMASK[7:0] back to `8'b11111111`. Separate asynchronous RX PCS reset ports are also available for each component: RXEYESCANRESET, RXFECRESET, RXPCSRESET, and RXPRBSCSCNTRST.

All asynchronous reset ports described in this section initiate the internal reset state machine when driven High. The internal reset state machines are held in reset state until these same reset ports are driven low. The guidelines for the pulse width of these asynchronous resets is one period of the reference clock, unless otherwise noted.

***Note*:** Do not use reset ports for the purpose of power down. For details on proper power-down usage, refer to Power Down.

## Resetting Multiple Lanes and Quads

Resetting multiple lanes in a Dual or multiple Duals affects the power supply regulation circuit (see Power Up/Down and Reset on Multiple Lanes).

## Reset Modes

The GTM transceiver TX and RX resets can operate in two different modes: sequential mode and single mode.

- **Sequential mode:** The reset state machine starts with an initialization or component reset input driven High and proceeds through all states after the requested reset states in the reset state machine, as shown in Figure 16 for the GTM transceiver TX or Figure 20 for the GTM transceiver RX until completion. The completion of sequential mode reset flow is signaled when (TX/RX)RESETDONE transitions from Low to High.
- **Single mode:** The reset state machine only executes the requested component reset independently for a predetermined time set by its attribute. The resets are executed through the corresponding asynchronous component reset port: TXCKALRESET, TXFECRESET, TXPCSRESET, TXPMARESET, RXEYESCANRESET, RXFECRESET, RXPCSRESET, and RXPRBSCSCNTRST. RX PMA single mode reset is not supported.

The GTM transceiver initialization reset is set to be in sequential mode. All component resets can be operated in either sequential mode or single mode. Reset modes have no impact on LCPLL reset. During normal operation, the GTM transceiver TX or GTM transceiver RX can be reset by applications in either sequential mode or single mode which provides flexibility to reset a portion of the GTM transceiver.

*Table 11:* **GTM Transceiver Reset Modes Ports**

| Port | Direction | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_RXRESETMODE[1:0] | In | Async | Reserved. Tie to `2'b01`. |
| CH[0/1]_TXRESETMODE[1:0] | In | Async | Reserved. Tie to `2'b00`. |

## LCPLL Reset

The LCPLL must be reset before it can be used. Each GTM transceiver dual has dedicated reset ports for its LCPLL. As shown in the following figure, PLLRESET is an input that resets LCPLL. PLLLOCK is an output that indicates the reset process is done. The guideline for asynchronous PLLRESET pulse width is one period of the reference clock. After a PLLRESET pulse, the internal reset controller generates an internal LCPLL reset followed by an internal SDM reset. The time required for LCPLL lock is affected by a few factors, such as bandwidth setting and clock frequency.

*Figure 15:* **LCPLL Reset Timing Diagram**

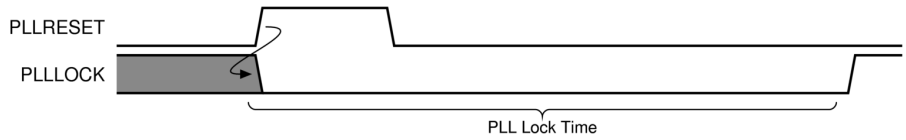



*Table 12:* **LCPLL Reset Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| PLLRESET | In | Async | Active-High signal that resets the LCPLL. |
| PLLRESETMASK[1:0] | In | Async | Reserved. Tied to `2'b11`. |
| PLLRESETBYPASSMODE | In | Async | Reserved. Tied Low. |
| PLLLOCK | Out | Async | Active-High LCPLL frequency lock signal indicates that the LCPLL frequency is within a predetermined tolerance. The GTM transceiver and its clock outputs are not reliable until this condition is met. |

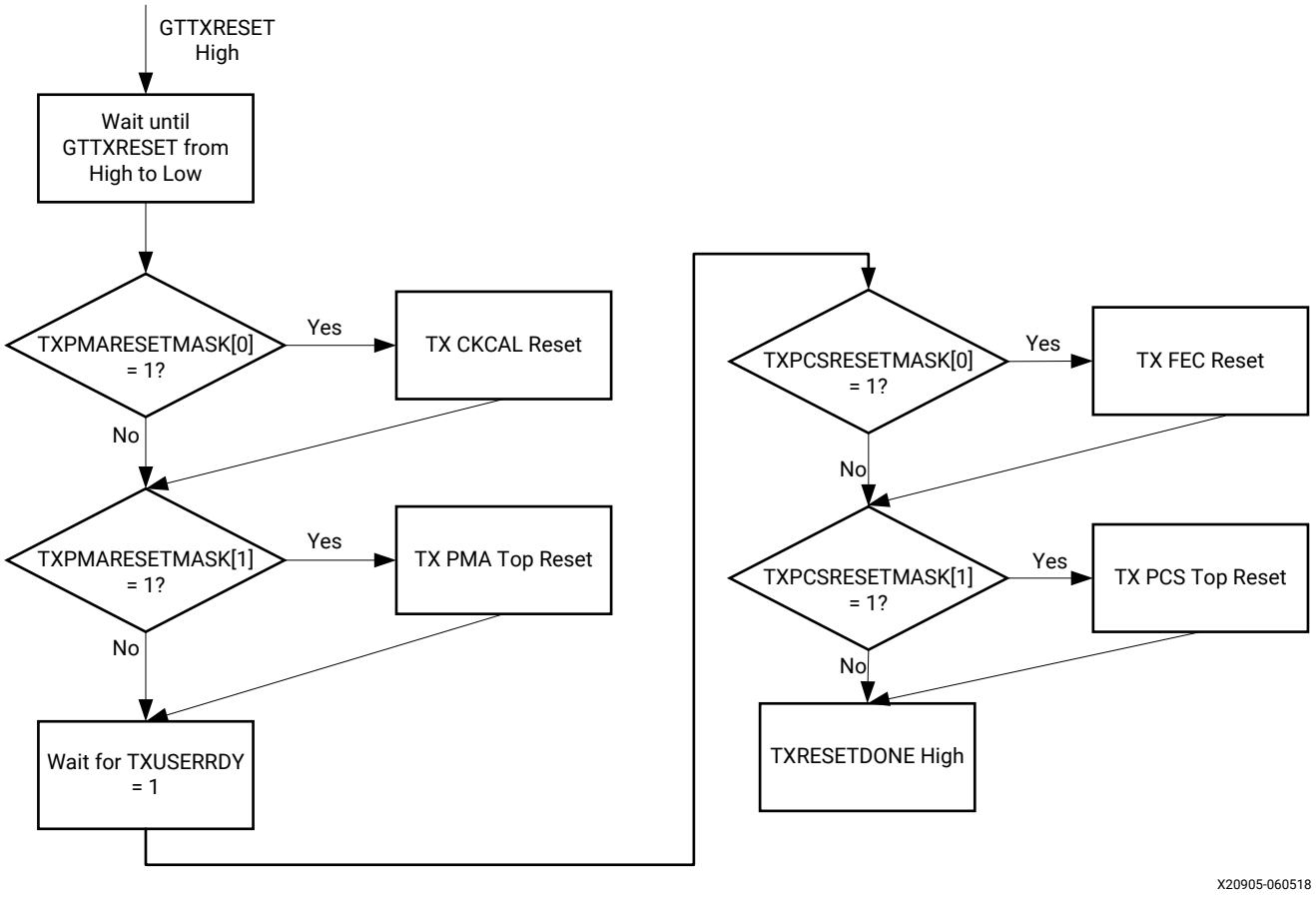
## TX Initialization and Reset

The GTM transceiver TX uses a reset state machine to control the reset process. The GTM transceiver TX is partitioned into two reset regions: TX PMA and TX PCS. The partition allows TX initialization and reset to be operated only in sequential mode, as shown in the figure below.

The initializing TX must use GTTXRESET in sequential mode. Activating the GTTXRESET input can automatically trigger a full asynchronous TX reset. The reset state machine executes the reset sequence, as shown in the figure below, covering the whole TX PMA and TX PCS. During normal operation, when needed, sequential mode allows you to reset the TX from activating TXPMARESET and continue the reset state machine until TXRESETDONE transitions from Low to High.

The TX reset state machine does not reset the PCS until TXUSERRDY is detected High. Drive TXUSERRDY High after all clocks used by the application including TXUSRCLK are shown as stable.

*Figure 16:* **GTM Transceiver TX Reset State Machine Sequence**




## Ports and Attributes

The table below lists ports required by the TX initialization process.

*Table 13:* **TX Initialization and Reset Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_GTTXRESET | In | Async | This port is driven High and then deasserted to start a TX reset sequence. The components to be reset are determined by TXPMARESETMASK and TXPCSRESETMASK. The resets are always performed sequentially. |
| CH[0/1]_TXRESETMODE[1:0] | In | Async | Reserved. Tie to `2'b00`. |
| CH[0/1]_TXPMARESETMASK[1:0] | In | Async | TX PMA reset mask selection:<br>Bit 1: TX PMA.<br>Bit 0: CKCAL. |

*Table 13:* **TX Initialization and Reset Ports** *(cont'd)*

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_TXPCSRESETMASK[1:0] | In | Async | TX PCS reset mask selection<br>Bit 1: TX PCS.<br>Bit 0: TX FEC. |
| CH[0/1]_TXUSERRDY | In | Async | This port is driven High by the user application when TXUSRCLK is stable. |
| CH[0/1]_TXPMARESETDONE | Out | Async | This active-High signal indicates TX PMA reset is complete. |
| CH[0/1]_TXRESETDONE | Out | TXUSRCLK | This active-High signal indicates the GTM transceiver TX has finished reset and is ready for use. This port is driven Low when GTTXRESET goes High and is not driven High until the GTM transceiver has completed all TX reset steps. |
| CH[0/1]_ TXCKALRESET | In | Async | This port is driven High and then deasserted to start a single mode reset on TX CKCAL. The reset is not dependent on TXPMARESETMASK setting. |
| CH[0/1]_ TXPMARESET | In | Async | This port is driven High and then deasserted to start a single mode reset on TX PMA. The reset is not dependent on TXPMARESETMASK setting. |
| CH[0/1]_ TXFECRESET | In | Async | This port is driven High and then deasserted to start a single mode reset on TX FEC. The reset is not dependent on TXPCSRESETMASK setting. |
| CH[0/1]_ TXPCSRESET | In | Async | This port is driven High and then deasserted to start a single mode reset on TX PCS. The reset is not dependent on TXPCSRESETMASK setting. |

The following table lists attributes required by GTM transceiver TX initialization. In general cases, the reset time required by the TX PMA or the TX PCS varies depending on line rate. The factors affecting PMA reset time and PCS reset time are the user-configurable attributes TX_PMA_RESET_TIME, TX_PCS_RESET_TIME, TX_CKCAL_RESET_TIME, and TX_FEC_RESET_TIME.

*Table 14:* **TX Initialization and Reset Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RST_TIME_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| TX_PCS_RESET_TIME | [14:10] | Represents the time duration to apply a TX PCS reset. Use the recommended value from the Wizard. Must be a non-zero value when TXPCSRESETMASK[1] is High and GTTXRESET initiates the reset process. |
| TX_PMA_RESET_TIME | [9:5] | Represents the time duration to apply a TX PMA reset. Use the recommended value from the Wizard. Must be a non-zero value when TXPMARESETMASK[1] is High and GTTXRESET initiates the reset process. |
| TX_CKCAL_RESET_TIME | [4:0] | Represents the time duration to apply a TX CLKGEN reset. Use the recommended value from the Wizard. Must be a non-zero value when TXPMARESETMASK[0] is High and GTTXRESET initiates the reset process. |

*Table 14:* **TX Initialization and Reset Attributes** *(cont'd)*

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RST_TIME_CFG1 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| TX_FEC_RESET_TIME | [4:0] | Represents the time duration to apply a TX CKCAL reset. Use the recommended value from the Wizard. Must be a non-zero value when TXPMARESETMASK[0] is High and GTTXRESET initiates the reset process. |
| CH[0/1]_RST_LP_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| TX_PCS_RESET_LOOP_ID | [11:8] | Reserved. Use the recommended value from the Wizard. |
| TX_PMA_RESET_LOOP_ID | [7:4] | Reserved. Use the recommended value from the Wizard. |
| TX_CKCAL_RESET_LOOP_ID | [3:0] | Reserved. Use the recommended value from the Wizard. |
| TX_FEC_RESET_LOOP_ID | [15:12] | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_RST_LP_ID_CFG1 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| TX_PCS_LOOPER_END_ID | [15:12] | Reserved. Use the recommended value from the Wizard. |
| TX_PCS_LOOPER_START_ID | [11:8] | Reserved. Use the recommended value from the Wizard. |
| TX_PMA_LOOPER_END_ID | [7:4] | Reserved. Use the recommended value from the Wizard. |
| TX_PMA_LOOPER_START_ID | [3:0] | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_RST_LP_CFG4 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| BYP_HDSHK_TX_PCS_RESET_LOOP | [3] | Reserved. Use the recommended value from the Wizard. |
| BYP_HDSHK_TX_CKCAL_RESET_LOOP | [0] | Reserved. Use the recommended value from the Wizard. |

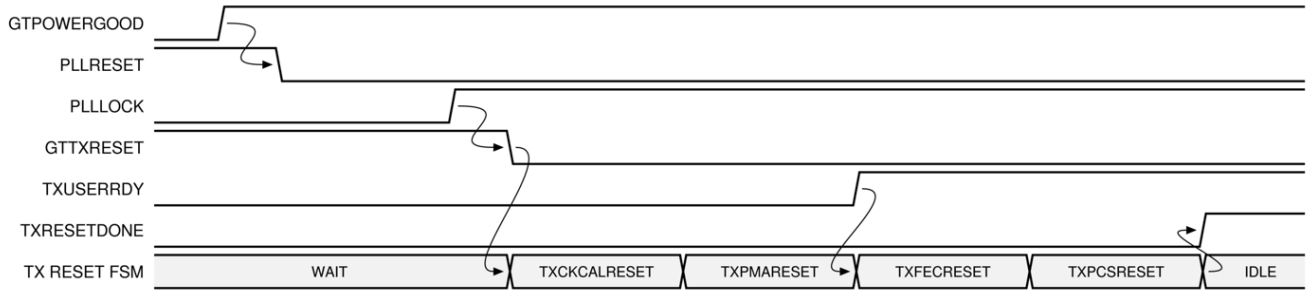
# GTM Transceiver TX Reset in Response to Completion of Configuration

The TX reset sequence shown in TX Initialization and Reset is not automatically started to follow global GSR. It must meet these conditions:

1. GTTXRESET must be used.
2. All TXPMARESETMASK and TXPCSRESETMASK bits should be set to High.
3. GTTXRESET cannot be driven Low until the associated PLL is locked.
4. Ensure that GTPOWERGOOD is High before releasing PLLRESET and GTTXRESET.

**RECOMMENDED:** *Use the associated PLLLOCK from the PLL to release GTTXRESET from High to Low as shown in the figure. The TX reset state machine waits when GTTXRESET is detected High and starts the reset sequence until GTTXRESET is released Low.*

*Figure 17:* **GTM Transmitter Initialization after Configuration**

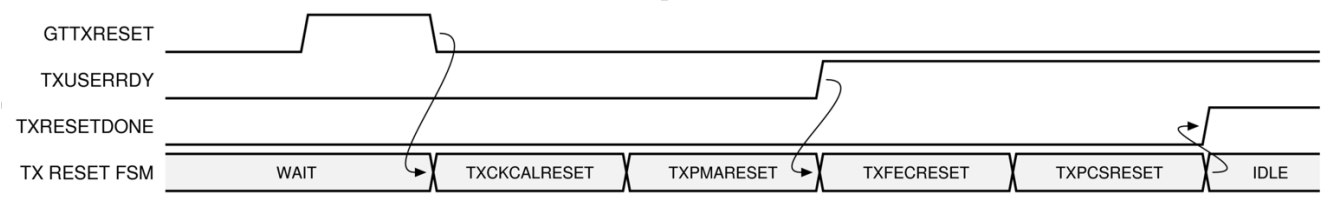



## GTM Transceiver TX Reset in Response to GTTXRESET Pulse in Full Sequential Reset

The GTM transceiver allows you to reset the entire TX completely at any time by sending GTTXRESET an active-High pulse. These conditions must be met when using GTTXRESET:

1. All TXPMARESETMASK and TXPCSRESETMASK bits should be held High during the reset sequence before TXRESETDONE is detected High.
2. The associated PLL must indicate locked.
3. The guideline for this asynchronous GTTXRESET pulse width is one period of the reference clock.

*Figure 18:* **GTM Transmitter Reset after GTTXRESET Pulse in Full Sequential Reset**




## GTM Transceiver TX Component Reset

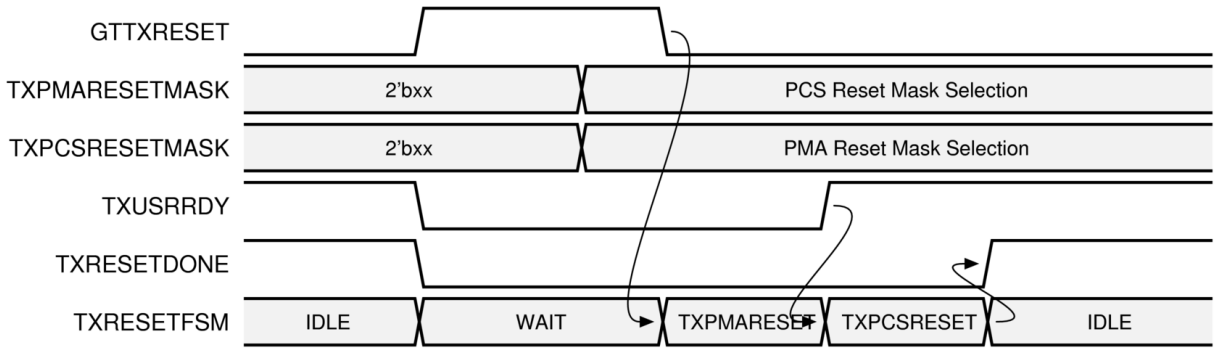
TX PMA and TX PCS can be reset individually. Component reset is enabled by setting the appropriate TXPMARESETMASK and TXPCSRESETMASK bits along with TXRESETMODE and then toggling GTTXRESET.

Driving GTTXRESET from High to Low starts the component reset process. All TXPMARESETMASK and TXPCSRESETMASK bits must be held constant during the reset process. The internal resets are toggled in sequence depending on TXPMARESETMASK and TXPCSRESETMASK selection.

If the TX PCS is to be reset, TXUSERRDY must be toggled High prior to the internal PCS reset signal being released, allowing TX reset to be completed.

Direct single-reset ports TXPMARESET, TXCKALRESET, TXPCSRESET, and TXFECRESET are available to perform single resets of the respective TX components. When direct single-reset ports are toggled, a single reset is performed regardless of TXPMARESETMASK and TXPCSRESETMASK selection. These ports must be held Low during any resets driven High by GTTXRESET.

*Figure 19:* **GTM Transmitter Reset after GTTXRESET Pulse in Component Sequential Mode**




The following table lists the recommended resets for common situations.

*Table 15:* **Recommended Transmitter Resets for Common Situations**

| Situation | Components to be Reset | Recommended TX Reset Setting | |
|---|---|---|---|
| | | TXPMARESETMASK | TXPCSRESETMASK[1] |
| After power up and configuration | PLL, Entire TX | 2'b11 | 2'b11 |
| After turning on a reference clock to the PLL being used | PLL, Entire TX | 2'b11 | 2'b11 |
| After changing the reference clock to the PLL being used | PLL, Entire TX | 2'b11 | 2'b11 |
| After assertion/deassertion of PLLPD for the PLL being used | PLL, Entire TX | 2'b11 | 2'b11 |
| After assertion/deassertion of TXPD | Entire TX | 2'b11 | 2'b11 |
| TX rate change | Entire TX | 2'b11 | 2'b11 |
| TX parallel clock source reset | TX PCS | 2'b00 | 2'b11 |

**Notes:**

1. TXPCSRESETMASK[0] can be set to 0 if the FEC is bypassed.

## After Power-up and Configuration

The PLL being used and the entire GTM TX require a reset after configuration. See GTM Transceiver TX Reset in Response to Completion of Configuration.

## After Turning on a Reference Clock to the LCPLL/RPLL Being Used

If the reference clock(s) changes or the GTM transceiver(s) are powered up after configuration, perform a full TX sequential reset after the PLL fully completes its reset procedure.

## After Changing the Reference Clock to the PLL being Used

Whenever the reference clock input to the PLL is changed, the PLL must be reset afterward to ensure that it locks to the new frequency. Perform a full TX sequential reset after the PLL fully completes its reset procedure.

## After Assertion/Deassertion of PLLPD for the PLL being Used

When the PLL being used goes back to normal operation after power down, the PLL must be reset. Perform a full TX sequential reset after the PLL fully completes its reset procedure.

## After Assertion/Deassertion of TXPD[1:0]

After the TXPD signal is deasserted, perform a full TX sequential reset.

## TX Rate Change

When a rate change is performed, a full TX sequential reset is required after the rate attributes have been updated.

## TX Parallel Clock Source Reset

The clocks driving TXUSRCLK must be stable for correct operation. Perform a TX PCS reset after the clock source re-locks.

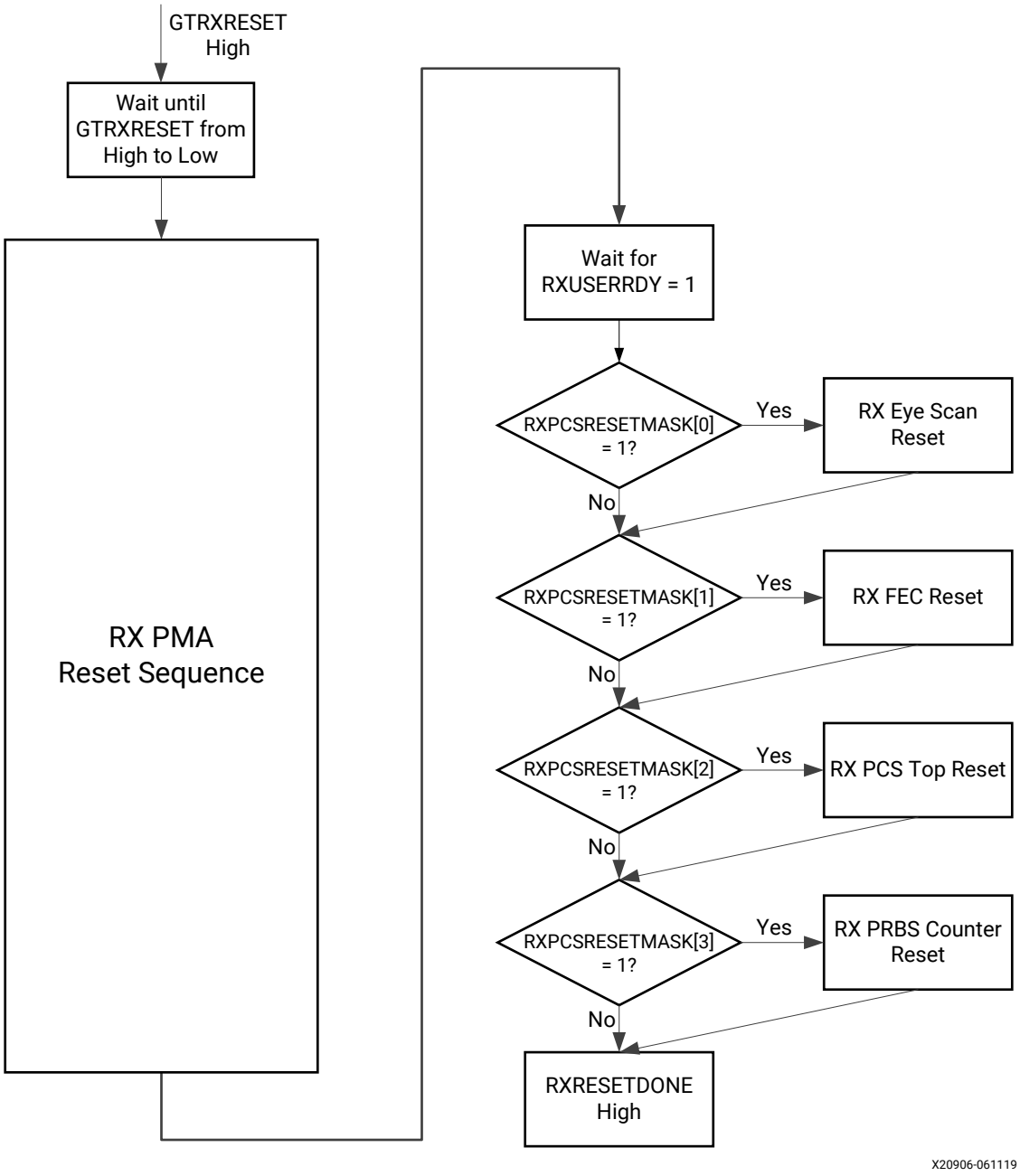
# RX Initialization and Reset

The GTM transceiver RX uses a reset state machine to control the reset process. Due to its complexity, the GTM transceiver RX is partitioned into more reset regions than the GTM transceiver TX. The partition allows RX initialization and reset to be operated in either sequential mode, as shown in the following figure, or single mode:

- **RX in Sequential Mode:** For GTM transceiver RX initialization reset and RX PMA reset, the reset must be executed using the GT Wizard IP. For RX PCS sequential reset, the components that are required to be reset are determined by setting the appropriate RXPCSRESETMASK bits to High. The reset sequence is then triggered by toggling GTRXRESET and then internal components resets are triggered sequentially. For proper RX PCS component reset, the user must set RXPMARESETMASK[7:0] to `8'b00000000` before toggling GTRXRESET. After RX PCS component reset, the user must set RXPMARESETMASK[7:0] back to `8'b11111111`. The reset state machine executes the reset sequence as shown in the following figure, covering the entire RX PCS datapath. During normal operation, the state machine runs until RXRESETDONE transitions from Low to High.
- **RX in Single Mode:** For the GTM transceiver RX, PMA single mode reset is not supported. For RX PCS, single mode reset can be executed using the asynchronous reset ports for each component: RXEYESCANRESET, RXFECRESET, RXPCSRESET, and RXPRBSCSCNTRST.

In sequential mode, the RX reset state machine does not reset the PCS until RXUSERRDY goes High. Drive RXUSERRDY High after all clocks used by the application, including RXUSRCLK, are shown to be stable.

*Figure 20:* **GTM Transceiver RX Reset State Machine Sequence**




## Ports and Attributes

The following table lists the ports required by the GTM transceiver RX initialization process.

*Table 16:* **RX Initialization and Reset Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_GTRXRESET | In | Async | This port is driven High and then deasserted to start a RX PCS reset sequence. The RX PCS components to be reset are determined by RXPCSRESETMASK. The resets are performed sequentially. For RX PMA reset, the user must go through the GT Wizard IP. |
| CH[0/1]_RXRESETMODE[1:0] | In | Async | Reserved. Tie to `2'b01`. |
| CH[0/1]_RXPMARESETMASK[7:0] | In | Async | Set to `8'b11111111` during normal operation. For RX PCS sequential reset, this port must be set to `8'b00000000` prior to toggling GTRXRESET. Immediately after RX PCS reset, this port must be set back to `8'b11111111`. |
| CH[0/1]_RXPCSRESETMASK[3:0] | In | Async | RX PCS reset mask selection<br>Bit 3: PRBS counter.<br>Bit 2: RX PCS top.<br>Bit 1: FEC.<br>Bit 0: Reserved. |
| CH[0/1]_RXUSERRDY | In | Async | This port is driven High by the user application when RXUSRCLK is stable. |
| CH[0/1]_RXPMARESETDONE | Out | Async | This active-High signal indicates RX PMA reset is complete. |
| CH[0/1]_RXRESETDONE | Out | RXUSRCLK | This active-High signal indicates the GTM transceiver RX has finished reset and is ready for use. This port is driven Low during RX initialization reset triggered through the GT Wizard IP, as well as when GTRXRESET goes High for RX PCS reset. This port is not driven High until the GTM transceiver has completed all RX reset steps. |
| CH[0/1]_RXADAPTRESET | In | Async | Reserved. Tie to `1'b0`. |
| CH[0/1]_RXADCCALRESET | In | Async | Reserved. Tie to `1'b0`. |
| CH[0/1]_RXADCCLKGENRESET | In | Async | Reserved. Tie to `1'b0`. |
| CH[0/1]_RXBUFRESET | In | Async | Reserved. Tie to `1'b0`. |
| CH[0/1]_RXCDRFRRESET | In | Async | Reserved. Tie to `1'b0`. |
| CH[0/1]_RXCDRPHRESET | In | Async | Reserved. Tie to `1'b0`. |
| CH[0/1]_RXDFERESET | In | Async | Reserved. Tie to `1'b0`. |
| CH[0/1]_RXDSPRESET | In | Async | Reserved. Tie to `1'b0`. |
| CH[0/1]_RXEYESCANRESET | In | Async | Reserved. Tie to `1'b0`. |
| CH[0/1]_RXFECRESET | In | Async | This port is driven High and then deasserted to start a single mode reset on the FEC. The reset is not dependent on RXPCSRESETMASK setting. |
| CH[0/1]_RXPCSRESET | In | Async | This port is driven High and then deasserted to start a single mode reset on the PCS top. The reset is not dependent on RXPCSRESETMASK setting. |
| CH[0/1]_RXPMARESET | In | Async | Reserved. Tie to `1'b0`. |
| CH[0/1]_RXPRBSCNTRESET | In | Async | This port is driven High and then deasserted to start a single mode reset on the PRBS counter. The reset is not dependent on RXPCSRESETMASK setting. |

The following table lists attributes required by GTM transceiver RX initialization. In general cases, the reset time required by the RX PMA or the RX PCS varies depending on line rate. The factors affecting PMA reset time and PCS reset time are the user-configurable attributes RX_PMA_RESET_TIME, RX_DFE_RESET_TIME, RX_ADAPT_RESET_TIME, RX_DSP_RESET_TIME, RX_ADC_CLKGEN_RESET_TIME, RX_CDRFREQ_RESET_TIME, RX_CDRPHASE_RESET_TIME, RX_PCS_RESET_TIME, RX_FEC_RESET_TIME, RX_PRBS_RESET_TIME and RX_EYESCAN_RESET_TIME.

*Table 17:* **RX Initialization and Reset Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RST_TIME_CFG2 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Bit Field** | **Description** |
| RX_ADAPT_RESET_TIME | [14:10] | Reserved. Represents the time duration to apply an RX adapt reset. Use the recommended value from the Wizard. |
| RX_DSP_RESET_TIME | [9:5] | Reserved. Represents the time duration to apply an RX DSP reset. Use the recommended value from the Wizard. |
| RX_ADC_CLKGEN_RESET_TIME | [4:0] | Reserved. Represents the time duration to apply an RX ADC CLKGEN reset. Use the recommended value from the Wizard. |
| CH[0/1]_RST_TIME_CFG3 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Bit Field** | **Description** |
| RX_CDRFREQ_RESET_TIME | [14:10] | Reserved. Represents the time duration to apply an RX CDR FR reset. Use the recommended value from the Wizard. |
| RX_CDRPHASE_RESET_TIME | [9:5] | Reserved. Represents the time duration to apply an RX CDR PH reset. Use the recommended value from the Wizard. |
| CH[0/1]_RST_TIME_CFG1 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Bit Field** | **Description** |
| RX_DFE_RESET_TIME | [14:10] | Reserved. Represents the time duration to apply an RX DFE reset. Use the recommended value from the Wizard. |
| RX_PMA_RESET_TIME | [9:5] | Reserved. Represents the time duration to apply an RX PMA reset. Use the recommended value from the Wizard. |
| CH[0/1]_RST_TIME_CFG4 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Bit Field** | **Description** |
| RX_PCS_RESET_TIME | [14:10] | Reserved. Represents the time duration to apply an RX PCS reset. Use the recommended value from the Wizard. Must be a non-zero value when RXPCSRESETMASK[2] is High and GTRXRESET initiates the reset process. |

*Table 17:* **RX Initialization and Reset Attributes** *(cont'd)*

| Attribute | Type | Description |
|---|---|---|
| RX_FEC_RESET_TIME | [9:5] | Reserved. Represents the time duration to apply an RX buffer reset. Use the recommended value from the Wizard. Must be a non-zero value when RXPCSRESETMASK[1] is High and GTRXRESET initiates the reset process. |
| RX_EYESCAN_RESET_TIME | [4:0] | Reserved. Represents the time duration to apply an eye scan reset. Use the recommended value from the Wizard. Must be a non-zero value when RXPCSRESETMASK[0] is High and GTRXRESET initiates the reset process. |
| CH[0/1]_RST_TIME_CFG5 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Bit Field** | **Description** |
| RX_PRBS_RESET_TIME | [4:0] | Reserved. Represents the time duration to apply a PRBS counter reset. Use the recommended value from the Wizard. Must be a non-zero value when RXPCSRESETMASK[3] is High and GTRXRESET initiates the reset process. |
| CH[0/1]_RST_LP_CFG1 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Bit Field** | **Description** |
| RX_DSP_RESET_LOOP_ID | [15:12] | Reserved. Use the recommended value from the Wizard. |
| RX_ADC_CLKGEN_RESET_LOOP_ID | [11:8] | Reserved. Use the recommended value from the Wizard. |
| RX_DFE_RESET_LOOP_ID | [7:4] | Reserved. Use the recommended value from the Wizard. |
| RX_PMA_RESET_LOOP_ID | [3:0] | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_RST_LP_CFG2 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Bit Field** | **Description** |
| RX_CDRFREQ_RESET_LOOP_ID | [15:12] | Reserved. Use the recommended value from the Wizard. |
| RX_CDRPHASE_RESET_LOOP_ID | [11:8] | Reserved. Use the recommended value from the Wizard. |
| RX_ADAPT_RESET_LOOP_ID | [3:0] | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_RST_LP_CFG3 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Bit Field** | **Description** |
| RX_PRBS_RESET_LOOP_ID | [15:12] | Reserved. Use the recommended value from the Wizard. |
| RX_PCS_RESET_LOOP_ID | [11:8] | Reserved. Use the recommended value from the Wizard. |
| RX_FEC_RESET_LOOP_ID | [7:4] | Reserved. Use the recommended value from the Wizard. |
| RX_EYESCAN_RESET_LOOP_ID | [3:0] | Reserved. Use the recommended value from the Wizard. |

*Table 17:* **RX Initialization and Reset Attributes** *(cont'd)*

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RST_LP_ID_CFG0 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Bit Field** | **Description** |
| RX_PCS_LOOPER_END_ID | [15:12] | Reserved. Use the recommended value from the Wizard. |
| RX_PCS_LOOPER_START_ID | [11:8] | Reserved. Use the recommended value from the Wizard. |
| RX_PMA_LOOPER_END_ID | [7:4] | Reserved. Use the recommended value from the Wizard. |
| RX_PMA_LOOPER_START_ID | [3:0] | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_RST_LP_CFG4 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| **Bit Name** | **Bit Field** | **Description** |
| BYP_HDSHK_RX_PCS_RESET_LOOP | [4] | Reserved. Use the recommended value from the Wizard. |
| BYP_HDSHK_TX_ADAPT_RESET_LOOP | [1] | Reserved. Use the recommended value from the Wizard. |

# GTM Transceiver RX Reset in Response to Completion of Configuration

The RX reset sequence shown in Figure 20 is not automatically started to follow global GSR. The GTM RX initialization reset must be executed through the GT Wizard IP. See*Virtex UltraScale+ FPGAs GTM Transceivers Wizard LogiCORE IP Product Guide* (PG315).

# GTM Transceiver RX PMA Reset

The GTM RX PMA reset must be executed through the GT Wizard IP. See*Virtex UltraScale+ FPGAs GTM Transceivers Wizard LogiCORE IP Product Guide* (PG315).

# GTM Transceiver RX PCS Reset in Response to GTRXRESET Pulse in Full Sequential Mode

The GTM transceiver allows you to reset the RX PCS at any time by sending GTRXRESET an active-High pulse. These conditions must be met when using GTRXRESET:

1. RXPMARESETMASK[7:0] should be set to `8'b00000000` prior to GTRXRESET being set High.
2. All RXPCSRESETMASK bits should be held High during the reset sequence before RXRESETDONE is detected High.

3. The associated PLL must indicate locked.
4. The guideline for this asynchronous GTRXRESET pulse width is one period of the reference clock.
5. After RXRESETDONE goes High, indicating RX PCS reset completion, RXPMARESETMASK[7:0] should be set back to `8'b11111111`.

*Figure 21:* **GTM Receiver Reset after GTRXRESET Pulse in Full Sequential Reset**


# GTM Transceiver RX PCS Component Reset

GTM transceiver RX component resets can be reset individually in either sequential mode or single mode. They are primarily used for special cases. These resets are needed when only a specific subsection needs to be reset.

Prior to driving GTRXRESET for RX PCS component reset, RXUSERRDY must be toggled High and RXPMARESETMASK[7:0] must be set to `8'b00000000`. Driving GTRXRESET from High to Low starts the component reset process. All RXPCSRESETMASK must be held constant during the reset process.

The port RXRESETDONE goes High after RX PCS component reset has been completed. After RXRESETDONE goes High, RXPMARESETMASK[7:0] should be set back to `8'b11111111`.

Direct single-reset ports RXPCSRESET, RXFECRESET, and RXPRBSCSCNTRST are available to perform single resets of the respective RX components. When direct single-reset ports are toggled, a single reset is performed regardless of RXPCSRESETMASK selection. These ports must be held Low during any sequential or single rests driven by GTRXRESET.

The table below lists the recommended receiver resets for common situations.

*Table 18:* **Recommended Receiver Resets for Common Situations**

| Situation | Components to be Reset | Recommended RX Reset Setting | | |
|---|---|---|---|---|
| | | GT Wizard IP RX Reset | RXPMARESETMASK | RXPCSRESETMASK[1] |
| After power up and configuration | PLL, Entire RX | Required | `8'b11111111` | `4'b1111` |
| After turning on a reference clock to the PLL being used | PLL, Entire RX | Required | `8'b11111111` | `4'b1111` |
| After changing the reference clock to the PLL being used | PLL, Entire RX | Required | `8'b11111111` | `4'b1111` |

*Table 18:* **Recommended Receiver Resets for Common Situations** *(cont'd)*

| Situation | Components to be Reset | Recommended RX Reset Setting | | |
|---|---|---|---|---|
| | | GT Wizard IP RX Reset | RXPMARESETMASK | RXPCSRESETMASK[1] |
| After assertion/deassertion of PLLPD for the PLL being used | PLL, Entire RX | Required | 8'b11111111 | 4'b1111 |
| After assertion/deassertion of RXPD | Entire RX | Required | 8'b11111111 | 4'b1111 |
| RX rate change | Entire RX | Required | 8'b11111111 | 4'b1111 |
| RX parallel clock source reset | RX PCS | N/A | 8'b00000000 | 4'b1111 |
| After remote power up | Entire RX | Required | 8'b11111111 | 4'b1111 |
| After connecting RXN/RXP | Entire RX | Required | 8'b11111111 | 4'b1111 |
| After recovered clock becomes stable | RX PCS | N/A | 8'b00000000 | 4'b1111 |
| After RX elastic buffer error | RX PCS | N/A | 8'b00000000 | 4'b1111 |
| After PRBS error | PRBS Error Counter | N/A | 8'b00000000 | 4'b1000 |

**Notes:**

1. RXPCSRESETMASK[1] can be set to 0 if the FEC is bypassed.

## After Power-up and Configuration

The PLL being used and the entire GTM RX require a reset after configuration. See GTM Transceiver RX Reset in Response to Completion of Configuration.

## After Turning on a Reference Clock to the PLL Being Used

If the reference clock(s) changes or GTM transceiver(s) are powered up after configuration, perform a full RX sequential reset after the PLL fully completes its reset procedure.

## After Changing the Reference Clock to the PLL Being Used

Whenever the reference clock input to the PLL is changed, the PLL must be reset afterward to ensure that it locks to the new frequency. Perform a full RX sequential reset after the PLL fully completes its reset procedure.

## After Assertion/Deassertion of PLLPD for the PLL Being Used

When the PLL being used goes back to normal operation after power down, the PLL must be reset. Perform a full RX sequential reset after the PLL fully completes its reset procedure.

## After Assertion/Deassertion of RXPD

After the RXPD signal is deasserted, perform a full RX initialization reset.

## RX Rate Change

When a rate change is performed, a full RX initialization reset is required after the rate attributes have been updated.

## RX Parallel Clock Source Reset

The clocks driving RXUSRCLK must be stable for correct operation. Perform an RX PCS reset after the clock source re-locks.

## After Remote Power-Up

If the source of incoming data is powered up after the GTM transceiver that is receiving its data has begun operating, a full RX initialization reset must be performed to ensure a clean lock to the incoming data.

## After Connecting RXN/RXP

When the RX data to the GTM transceiver comes from a connector that can be plugged in and unplugged, a full RX initialization reset must be performed when the data source is plugged in to ensure that it can lock to incoming data.

## After Recovered Clock Becomes Stable

Depending on the design of the clocking scheme, it is possible for the RX reset sequence to be completed before the CDR is locked to the incoming data. In this case, the recovered clock might not be stable when RXRESETDONE is asserted. When the RX buffer is used, a sequential component reset targeting the RX PCS must be triggered after the recovered clock becomes stable.

Refer to the UltraScale+ device data sheets (see http://www.xilinx.com/documentation) for successful CDR lock-to-data criteria.

## After an RX Elastic Buffer Error

After an RX elastic buffer overflow or underflow, a sequential component reset targeting the RX PCS must be triggered to ensure correct behavior.

## After a PRBS Error

PRBSCNTRESET is asserted to reset the PRBS error counter.

# Power Down

The GTM transceiver offers different levels of power control. Each channel in each direction can be powered down separately. The PLLPD port directly affects the LCPLL.

## Ports and Attributes

The following table defines the power-down ports.

*Table 19:* **Power-Down Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| PLLPD | In | Async | This active-High signal powers down the LCPLL. |

The following table defines the power-down attributes.

*Table 20:* **Power-Down Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_TX_ANA_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| TXPWRDN_B | [1:0] | Powers down channel TX:<br>`2'b00`: Power down.<br>`2'b11`: Power up. |
| RST_CFG | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| RX_PDB_CH0 | [2] | This active-Low signal powers down channel 0 RX. |
| RX_PDB_CH1 | [3] | This active-Low signal powers down channel 1 RX. |

## PLL Power Down

To activate the LCPLL power-down mode, the active-High PLLPD signal is asserted. When PLLPD is asserted, the LCPLL is powered down. As a result, all clocks derived from the PLL are stopped. Recovery from this power state is indicated by the PLL lock signal PLLLOCK.

## TX and RX Power Down

TX and RX power control signals can be used independently. Only two power states are supported, as shown in the following table. Powering up/down multiple lanes in a Dual or multiple Duals affects the power supply regulation circuit (see Power Up/Down and Reset on Multiple Lanes).

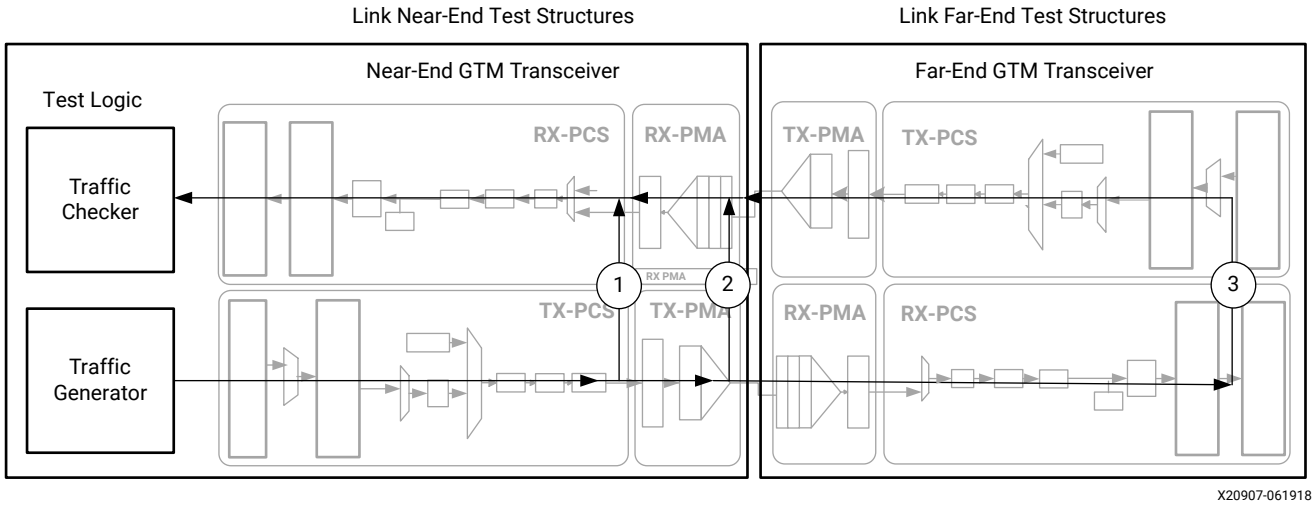
*Table 21:* **TX and RX Power Control Signals**

| Bit Name | Value | Description |
|---|---|---|
| TXPWRDN_B | 2'b11 | Normal mode. Transceiver TX is actively sending data. |
| TXPWRDN_B | 2'b00 | Power-down mode. Transceiver TX is idle. |
| [RX_PDB_CH0, RX_PDB_CH1] | 2'b11 | Normal mode. Transceiver RX for channel 0 and channel 1 are actively receiving data. |
| [RX_PDB_CH0, RX_PDB_CH1] | 2'b00 | Power-down mode. Transceiver RX for channel 0 and channel 1 are idle. |

# Loopback

Loopback modes are specialized configurations of the transceiver datapath where the traffic stream is folded back to the source. Typically, a specific pattern is transmitted then compared to check for errors. The following figure illustrates a loopback test configuration with three different loopback modes.

*Figure 22:* **Loopback Testing Overview**




Loopback test modes fall into two broad categories:

- Near-end loopback modes loop transmit data back in the transceiver closest to the traffic generator.
- Far-end loopback modes loop received data back in the transceiver at the far end of the data link.

Loopback testing can be used either during development or in deployed equipment for fault isolation. The traffic patterns used can be either application traffic patterns or specialized pseudo-random bit sequences. Each GTM transceiver has a built-in PRBS generator and checker.

Each GTM transceiver features several loopback modes to facilitate testing:

- Near-end PCS loopback (Path 1 in the above figure). While in near-end PCS loopback, the RX XCLK domain is clocked by the TX parallel clock (TX XCLK). To set the GTM transceiver in near-end PCS loopback, follow the sequence below:
  - Bypass GTM controller helper logic through the GT Wizard IP (signal ch0/1_resetsol_en[3:0]).
  - Set CH[0/1]_RXCDRHOLD = `1'b1`.
  - Reset the GTM channel through the GT Wizard IP (signal hb_gtwiz_reset_all_in).

  ***Note:*** For more information on the GT Wizard IP signals refer to the *Virtex UltraScale+ FPGAs GTM Transceivers Wizard LogiCORE IP Product Guide* (PG315).
- Near-end PMA loopback (path 2 in the above figure). The GT Wizard IP must be used to set the GTM transceiver in near-end PMA loopback. See *Virtex UltraScale+ FPGAs GTM Transceivers Wizard LogiCORE IP Product Guide* (PG315).

- Far-end PCS loopback (path 3 in the above figure). The transceiver in far-end PCS loopback must use the same reference clock used by the transceiver that is the source of the loopback data.

## Ports and Attributes

The following table defines the loopback ports.

*Table 22:* **Loopback Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_LOOPBACK[2:0] | In | Async | Loopback control for channel 0/1:<br>`3'b000`: Normal operation.<br>`3'b001`: Near-End PCS Loopback.<br>`3'b010`: Near-End PMA Loopback.<br>`3'b011`: Reserved.<br>`3'b100`: Reserved.<br>`3'b101`: Reserved.<br>`3'b110`: Far-End PCS Loopback. |

The following table defines the loopback attributes.

*Table 23:* **Loopback Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_TX_LPBK_CFG0 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_TX_LPBK_CFG1 | 16-bit | Reserved. Use the recommended value from the Wizard. |

# Dynamic Reconfiguration Port

The dynamic reconfiguration port (DRP) allows the dynamic change of parameters of the GTM_DUAL primitives. The DRP interface is a processor-friendly synchronous interface with an address bus (DRPADDR) and separate data buses for reading (DRPDO) and writing (DRPDI) configuration data to the primitives. An enable signal (DRPEN), a read/write signal (DRPWE), and a ready/valid signal (DRPRDY) are the control signals that implement read and write operations, indicate operation completion, or indicate the availability of data.

## Ports and Attributes

The following table shows the DRP related ports for the GTM_DUAL.

*Table 24:* **DRP Ports of GTM_DUAL**

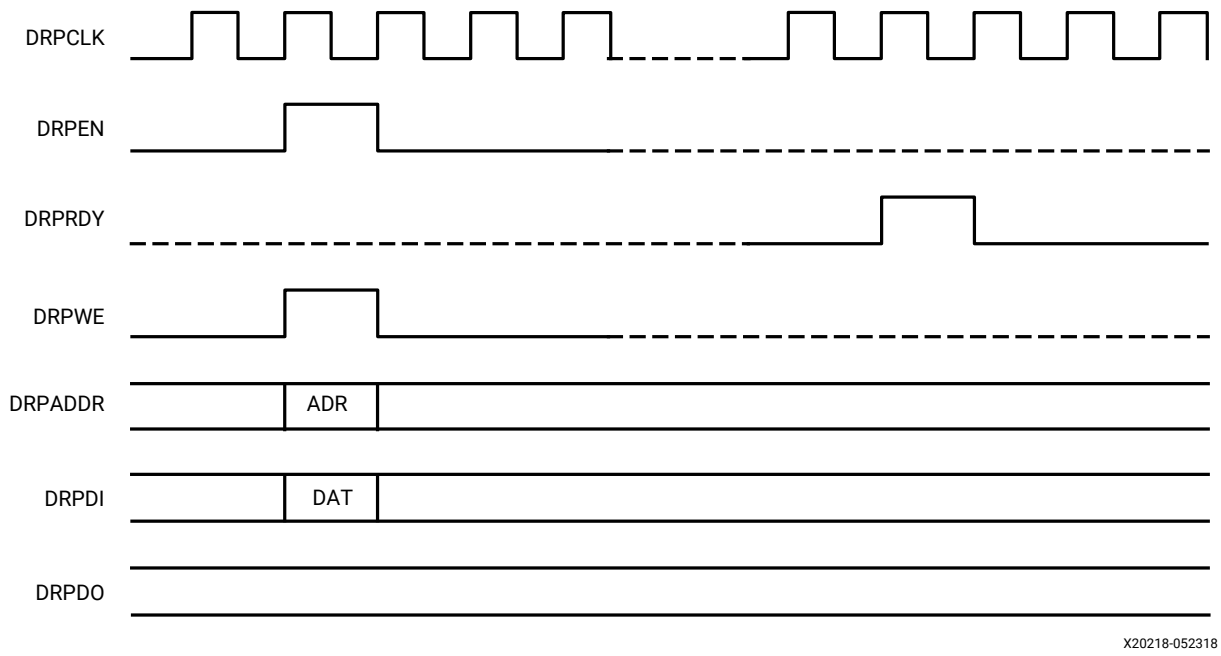
| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| DRPADDR[10:0] | In | DRPCLK | DRP address bus. |
| DRPCLK | In | N/A | DRP interface clock. |
| DRPEN | In | DRPCLK | DRP enable signal.<br>0: No read or write operation performed.<br>1: Enables a read or write operation.<br>For write operations. DRPWE and DRPEN should be driven High for one DRPCLK cycle only. See Figure 23 for correct operation. For read operations, DRPEN should be driven High for one DRPCLK cycle only. See Figure 24 for correct operation. |
| DRPDI[15:0] | In | DRPCLK | Data bus for writing configuration data from the interconnect logic resources to the transceivers. |
| DRPRDY | Out | DRPCLK | Indicates operation is complete for write operations and data is valid for read operations. If writing or reading a R/W register, DRPRDY asserts six DRPCLK cycles after initiating a DRP transaction. For read-only registers, the number of DRPCLK cycles for DRPRDY assertion depends on the relationship between the DRPCLK frequency and the USRCLK frequency. For read-only registers, if a DRPRDY is not seen within 500 DRPCLK cycles after initiating a DRP transaction, reset the DRP interface. |
| DRPDO[15:0] | Out | DRPCLK | Data bus for reading configuration data from the GTM transceiver to the interconnect logic resources. |
| DRPWE | In | DRPCLK | DRP write enable.<br>0: Read operation when DRPEN = 1.<br>1: Write operation when DRPEN is 1.<br>For write operations, DRPWE and DRPEN should be driven High for one DRPCLK cycle only. See Figure 23 for correct operation. |
| DRPRST | In | DRPCLK | DRP reset. Reading read-only registers while the XCLK is not toggling (e.g., during reset or change of reference clocks) causes the DRP to not return a DRPRDY signal and prevent further DRP transactions. In such an event, DRPRST must be pulsed to reset the DRP interface before initiating further DRP transactions. |

# Usage Model

## Write Operation

The following figure shows the DRP write operation timing. New DRP operations can be initiated when DRPRDY is asserted.

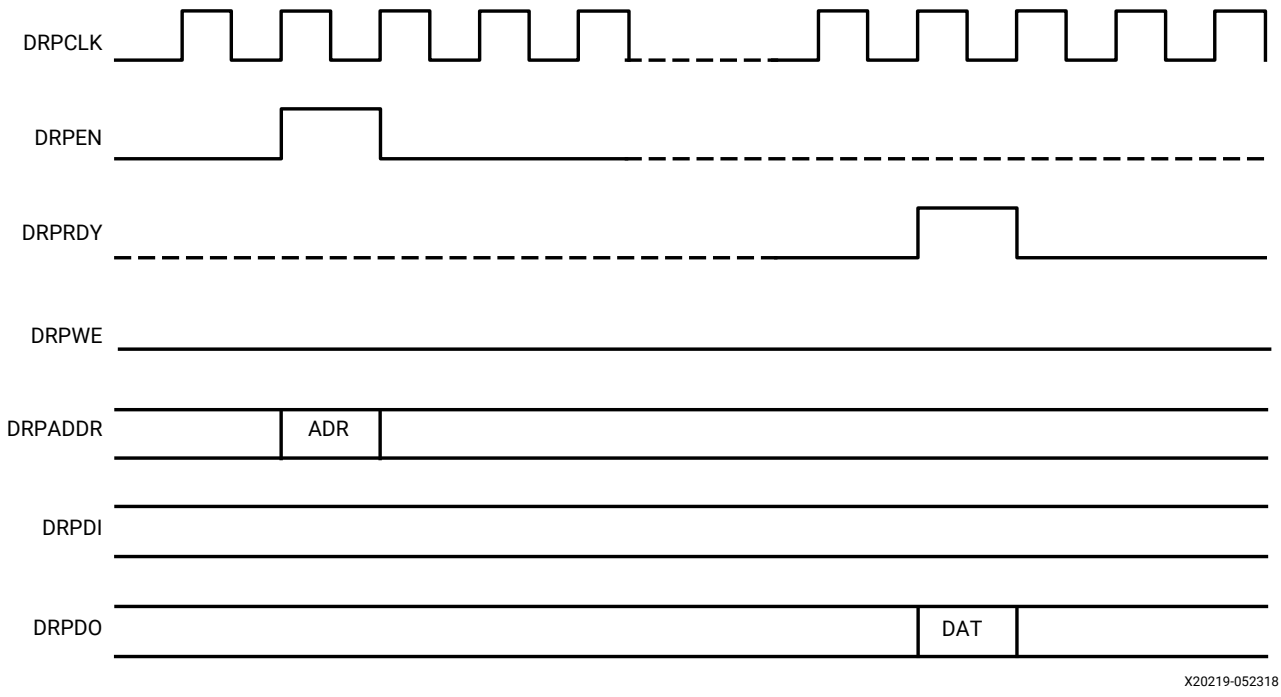
*Figure 23:* **DRP Write Timing**




### Read Operation

The following figure shows the DRP read operation timing. New DRP operations can be initiated when DRPY is asserted.

*Figure 24:* **DRP Read Timing**



# Digital Monitor

The receiver uses an adaptive algorithm in optimizing a link. The digital monitor provides visibility into the current state of these adaptation loops. Digital monitor requires a clock such as DRPCLK. CH0/1_RXUSRCLK2 can be used for this. The attributes CH[0/1]_RX_APT_CFG18A[15:12] and CH[0/1]_RX_DSP_CFG[7:3] select the adaptation loops monitored on the CH[0/1]_DMONITOROUT port. The output port CH[0/1]_DMONITOROUT contains the current code(s) for a selected loop. A loop has three steady states: min, max, or dithering.

## Ports and Attributes

The following table shows the GTM digital monitor ports.

*Table 25:* **Digital Monitor Ports**

| Port | Dir | Clock Domain | Description | | |
|---|---|---|---|---|---|
| CH[0/1]_DMONITOROUT[31:0] | Out | Async/Local Clock | Digital monitor output bus for channel 0/1. | | |
| | | | **Loop Select** | **Loop Select Value** | **Loop** |
| | | | DFELOOPSEL | 4'b0000 | [15:3] - H1_BINDAC |
| | | | | 4'b0001 | [30:25] - GC_BINDAC |
| | | | | | [15:10] - KH_BINDAC |
| | | | | 4'b0010 | [30:25] - KL_BINDAC |
| | | | | | [14:8] - OS_BINDAC |
| | | | FFELOOPSEL | 5'b00001 | [28:23] - FFE_HM4 |
| | | | | | [22:18] - FFE_HP10 |
| | | | | | [17:13] - FFE_HP09 |
| | | | | | [12:8] - FFE_HP08 |
| | | | | | [7:3] - FFE_HP07 |
| | | | | 5'b00010 | [31:26] - FFE_HP06 |
| | | | | | [25:20] - FFE_HP05 |
| | | | | | [19:13] - FFE_HP04 |
| | | | | | [12:6] - FFE_HP03 |
| | | | | 5'b00011 | [31:24] - FFE_HP02 |
| | | | | | [23:16] - FFE_HM01 |
| | | | | | [15:9] - FFE_HM02 |
| | | | | | [8:3] - FFE_HM03 |
| CH[0/1]_DMONITORCLK | In | Async | Channel 0/1 digital monitor clock. | | |
| CH[0/1]_DMONITORFIFORESET | In | Async | Reserved. Tie to GND. | | |
| CH[0/1]_DMONITOROUTCLK | Out | Async | Channel 0/1 internal clock from adaptation loops. | | |

*Table 25:* **Digital Monitor Ports** *(cont'd)*

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| DMONITOROUTPLLCLK | Out | Async | Internal TX calibration clock. |

The following table shows the GTM digital monitor attributes.

*Table 26:* **Digital Monitor Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RX_MON_CFG | 16-bit | Reserved. |
| **Bit Name** | **Bit Field** | **Description** |
| DMON_ENABLE | [0] | Enables digital monitor for channel 0/1. |
| DMON_SRC | [2:1] | Enables RX DMON path for channel 0/1. Must be set to `2'b00` when reading RX adaptation loops. |
| DMON_FIFOBYPASS | [3] | Enables RX DMON path for channel 0/1. Must set to `1'b0` when reading RX adaptation loops. |
| CH[0/1]_RX_APT_CFG8B | 16-bit | Reserved. |
| **Bit Name** | **Bit Field** | **Description** |
| CLKSEL | [15] | Set to `1'b0` to read DFE/FFE data. |
| DEMONCON | [14:12] | Selector for DMON data for channel 0/1.<br>`3'b100`: Choose DFE data through DFELOOPSEL.<br>`3'b110`: Choose FFE data through FFELOOPSEL. |
| CH[0/1]_RX_APT_CFG17B | 16-bit | Reserved. |
| **Bit Name** | **Bit Field** | **Description** |
| TESTEN | [13] | Set to `1'b1` when reading DFE adaptation loops. |

*Table 26:* **Digital Monitor Attributes** *(cont'd)*

| Attribute | Type | Description | | |
|---|---|---|---|---|
| CH[0/1]_RX_APT_CFG18A | 16-bit | Reserved. | | |
| **Bit Name** | **Bit Field** | **Description** | | |
| DFELOOPSEL | [15:12] | Selector to monitor DFE adaptation loops for channel 0/1. See output using CH[0/1]_DMONITOROUT port. | | |
| | | **DFELOOPSEL** | **DFE Loop** | **Code Mapping Range** |
| | | `4'b0000` | H1_BINDAC | Signed loop in twos complement format:<br>`13'h1000`: Min<br>`13'h0000`: Neutral<br>`13'h0111`: Max |
| | | `4'b0001` | GC_BINDAC | Magnitude:<br>`6'd0`: Min<br>`6'd63`: Max |
| | | | KH_BINDAC | Magnitude:<br>`6'd0`: Min<br>`6'd63`: Max |
| | | `4'b0010` | KL_BINDAC | Magnitude:<br>`6'd0`: Min<br>`6'd63`: Max |
| | | | OS_BINDAC | Magnitude:<br>`7'd0`: Min<br>`7'd64`: Neutral<br>`7'd127`: Max |
| CH[0/1]_RX_DSP_CFG | 16-bit | Reserved. | | |
| **Bit Name** | **Bit Field** | **Description** | | |
| TESTC | [13:12] | Set to `2'b10` when reading FFE adaptation loops. | | |

*Table 26:* **Digital Monitor Attributes** *(cont'd)*

| Attribute | Type | Description | | |
|---|---|---|---|---|
| FFELOOPSEL | [7:3] | Selector to monitor FFE adaptation loops for channel 0/1. | | |
| | | **FFELOOPSEL** | **DFE Loop** | **Code Mapping Range** |
| | | `5'b00001` | FFE_HM04 | Signed loop in twos complement format:<br>`6'b100000`: Min<br>`6'b000000`: Neutral<br>`6'b011111`: Max |
| | | | FFE_HP10 | Signed loop in twos complement format:<br>`5'b10000`: Min<br>`5'b00000`: Neutral<br>`5'b01111`: Max |
| | | | FFE_HP09 | Signed loop in twos complement format:<br>`5'b10000`: Min<br>`5'b00000`: Neutral<br>`5'b01111`: Max |
| | | | FFE_HP08 | Signed loop in twos complement format:<br>`5'b10000`: Min<br>`5'b00000`: Neutral<br>`5'b01111`: Max |
| | | | FFE_HP07 | Signed loop in twos complement format:<br>`5'b10000`: Min<br>`5'b00000`: Neutral<br>`5'b01111`: Max |

*Table 26:* **Digital Monitor Attributes** *(cont'd)*

| Attribute | Type | Description | | |
|---|---|---|---|---|
| FFELOOPSEL<br>*(Cont'd)* | [7:3] | 5'b00010 | FFE_HP06 | Signed loop in twos complement format:<br>6'b100000: Min<br>6'b000000: Neutral<br>6'b011111: Max |
| | | | FFE_HP05 | Signed loop in twos complement format:<br>6'b100000: Min<br>6'b000000: Neutral<br>6'b011111: Max |
| | | | FFE_HP04 | Signed loop in twos complement format:<br>7'b1000000: Min<br>7'b0000000: Neutral<br>7'b0111111: Max |
| | | | FFE_HP03 | Signed loop in twos complement format:<br>7'b1000000: Min<br>7'b0000000: Neutral<br>7'b0111111: Max |

*Table 26:* **Digital Monitor Attributes** *(cont'd)*

| Attribute | Type | Description | | |
|---|---|---|---|---|
| FFELOOPSEL *(Cont'd)* | [7:3] | 5'b00011 | FFE_HP02 | Signed loop in twos complement format:<br>8'b10000000: Min<br>8'b00000000: Neutral<br>8'b01111111: Max |
| | | | FFE_HM01 | Signed loop in twos complement format:<br>8'b10000000: Min<br>8'b00000000: Neutral<br>8'b01111111: Max |
| | | | FFE_HM02 | Signed loop in twos complement format:<br>7'b1000000: Min<br>7'b0000000: Neutral<br>7'b0111111: Max |
| | | | FFE_HM03 | Signed loop in twos complement format:<br>6'b100000: Min<br>6'b000000: Neutral<br>6'b011111: Max |

# Use Mode

Reading loop values out of Digital Monitor requires a clock on input port CH0/1_DMONITORCLK, change adaptation loop select through DRP, and monitor output CH0/1_DMONITOROUT. Set attributes DMON_ENABLE, DMON_SRC, DMON_FIFOBYPASS, TESTEN, DEMONCON, CLKSEL, TESTC, and DFELOOPSEL or FFELOOPSEL via the DRP port to enable the digital monitor and select the appropriate loop for monitoring. The DRP locations of the attributes are follows.

Channel 0:

- 0x082[0]: DMON_ENABLE
- 0x082[2:1]: DMON_SRC
- 0x082[3]: DMON_FIFOBYPASS
- 0x045[13]: TESTEN
- 0x033[14:12]: DEMONCON
- 0x033[15]: CLKSEL

- 0x05C[13:12]: TESTC
- 0x046[15:12]: DFELOOPSEL
- 0x05C[7:3]: FFELOOPSEL

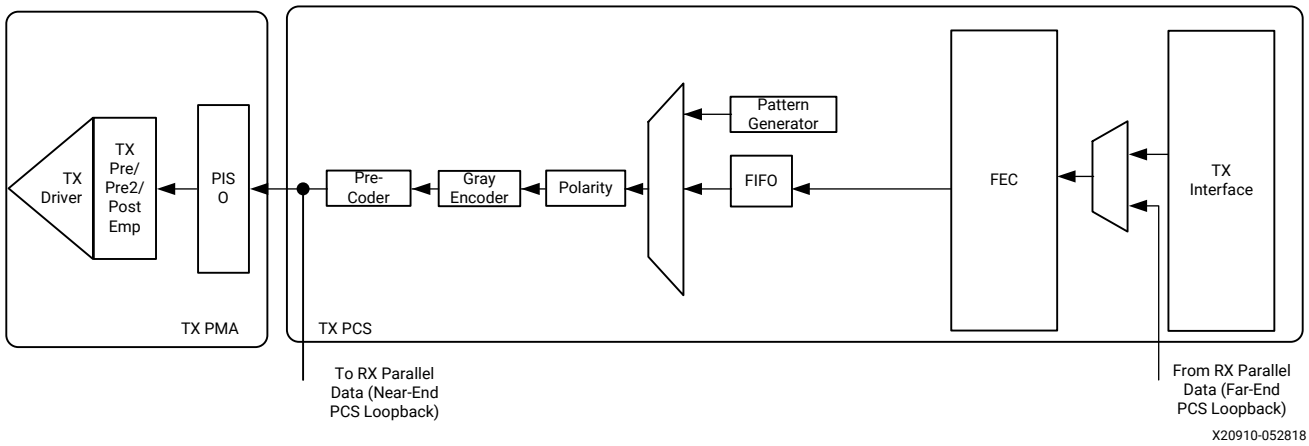
Channel 1:

- 0x282[0]: DMON_ENABLE
- 0x282[2:1]: DMON_SRC
- 0x282[3]: DMON_FIFOBYPASS
- 0x245[13]: TESTEN
- 0x233[14:12]: DEMONCON
- 0x233[15]: CLKSEL
- 0x25C[13:12]: TESTC
- 0x246[15:12]: DFELOOPSEL
- 0x25C[7:3]: FFELOOPSEL

# Transmitter

This chapter shows how to configure and use each of the functional blocks inside the transmitter (TX). Each transceiver includes an independent transmitter, which consist of a PCS and a PMA. The following figure shows the functional blocks of the transmitter. Parallel data flows from the device logic into the TX interface through the PCS and PMA, and then out the TX driver as high-speed serial data.

*Figure 25:* **GTM Transceiver TX Block Diagram**




The key elements within the GTM transceiver TX are:

1. TX Interface
2. TX FEC
3. TX Buffer
4. TX Pattern Generator
5. TX Polarity Control
6. TX Gray Encoder
7. TX Pre-Coder
8. TX Fabric Clock Output Control
9. TX Configurable Driver

# TX Interface

The TX interface is the gateway to the TX datapath of the GTM transceiver. Applications transmit data through the GTM transceiver by writing data to the TXDATA port on the positive edge of TXUSRCLK2. Port widths can be 64 and 128 bits for NRZ mode, or 80, 128, 160, and 256 bits for PAM4 mode. The rate of the parallel clock (TXUSRCLK2) at the interface is determined by the TX line rate and the width of the TXDATA port. A second parallel clock (TXUSRCLK) must be provided for the internal PCS logic in the transmitter. This section shows how to drive the parallel clocks and explains the constraints on those clocks for correct operation.

## Interface Width Configuration

The GTM transceiver contains a 64-bit internal datapath in NRZ mode and an 80-bit and 128-bit internal datapath in PAM4 mode that is configurable by setting the TX_INT_DATA_WIDTH attribute. When the FEC is enabled, only the 80-bit internal datapath may be used. The interface width is configurable by setting the TX_DATA_WIDTH attribute. In NRZ mode, TX_DATA_WIDTH can be configured to 64 or 128 bits. In PAM4 mode, TX_DATA_WIDTH can be configured to 80, 128, 160, or 256 bits. When the FEC is enabled, only the 80-bit or 160-bit data width can be selected.

The following table shows how the interface width for the TX datapath is selected.

*Table 27:* **TX Interface Datapath Configuration**

| Encoding | FEC Allowed? | TX_DATA_WIDTH Encoding | TX Data Width Selection | TX_INT_DATA_WIDTH Encoding | TX Internal Datapath Selection |
|---|---|---|---|---|---|
| NRZ | No | 0 | 64 | 0 | 64 |
| NRZ | No | 2 | 128 | 0 | 64 |
| PAM4 | Yes | 1 | 80 | 1 | 80 |
| PAM4 | Yes | 3 | 160 | 1 | 80 |
| PAM4 | No | 2 | 128 | 2 | 128 |
| PAM4 | No | 4 | 256 | 2 | 128 |

The following figure shows how the TX data is transmitted.

*Figure 26:* **TX Data Transmitted**

| | <<<Data Transmission Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data Transmitted | TXDATA[31:24] | | | | | | | | TXDATA[23:16] | | | | | | | | TXDATA[15:8] | | | | | | | | TXDATA[7:0] | | | | | | | |

| | <<<Data Transmission Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| Data Transmitted | TXDATA[63:56] | | | | | | | | TXDATA[55:48] | | | | | | | | TXDATA[47:40] | | | | | | | | TXDATA[39:32] | | | | | | | |

| | <<<Data Transmission Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 96 | 95 | 94 | 93 | 92 | 91 | 90 | 89 | 88 | 87 | 86 | 85 | 84 | 83 | 82 | 81 | 80 | 79 | 78 | 77 | 76 | 75 | 74 | 73 | 72 | 71 | 70 | 69 | 68 | 67 | 66 | 65 |
| Data Transmitted | TXDATA[96:89] | | | | | | | | TXDATA[88:81] | | | | | | | | TXDATA[80:73] | | | | | | | | TXDATA[72:65] | | | | | | | |

| | <<<Data Transmission Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 127 | 126 | 125 | 124 | 123 | 122 | 121 | 120 | 119 | 118 | 117 | 116 | 115 | 114 | 113 | 112 | 111 | 110 | 109 | 108 | 107 | 106 | 105 | 104 | 103 | 102 | 101 | 100 | 99 | 98 | 97 | 96 |
| Data Transmitted | TXDATA[127:120] | | | | | | | | TXDATA[119:112] | | | | | | | | TXDATA[111:104] | | | | | | | | TXDATA[103:96] | | | | | | | |

| | <<<Data Transmission Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 159 | 158 | 157 | 156 | 155 | 154 | 153 | 152 | 151 | 150 | 149 | 148 | 147 | 146 | 145 | 144 | 143 | 142 | 141 | 140 | 139 | 138 | 137 | 136 | 135 | 134 | 133 | 132 | 131 | 130 | 129 | 128 |
| Data Transmitted | TXDATA[159:152] | | | | | | | | TXDATA[151:144] | | | | | | | | TXDATA[143:136] | | | | | | | | TXDATA[135:128] | | | | | | | |

| | <<<Data Transmission Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 191 | 190 | 189 | 188 | 187 | 186 | 185 | 184 | 183 | 182 | 181 | 180 | 179 | 178 | 177 | 176 | 175 | 174 | 173 | 172 | 171 | 170 | 169 | 168 | 167 | 166 | 165 | 164 | 163 | 162 | 161 | 160 |
| Data Transmitted | TXDATA[191:184] | | | | | | | | TXDATA[183:176] | | | | | | | | TXDATA[175:168] | | | | | | | | TXDATA[167:160] | | | | | | | |

| | <<<Data Transmission Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 223 | 222 | 221 | 220 | 219 | 218 | 217 | 216 | 215 | 214 | 213 | 212 | 211 | 210 | 209 | 208 | 207 | 206 | 205 | 204 | 203 | 202 | 201 | 200 | 199 | 198 | 197 | 196 | 195 | 194 | 193 | 192 |
| Data Transmitted | TXDATA[223:216] | | | | | | | | TXDATA[215:208] | | | | | | | | TXDATA[207:200] | | | | | | | | TXDATA[199:192] | | | | | | | |

| | <<<Data Transmission Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 255 | 254 | 253 | 252 | 251 | 250 | 249 | 248 | 247 | 246 | 245 | 244 | 243 | 242 | 241 | 240 | 239 | 238 | 237 | 236 | 235 | 234 | 233 | 232 | 231 | 230 | 229 | 228 | 227 | 226 | 225 | 224 |
| Data Transmitted | TXDATA[255:248] | | | | | | | | TXDATA[247:240] | | | | | | | | TXDATA[239:232] | | | | | | | | TXDATA[231:224] | | | | | | | |

# TXUSRCLK and TXUSRCLK2 Generation

The TX interface includes two parallel clocks: TXUSRCLK and TXUSRCLK2. TXUSRCLK is the internal clock for the PCS logic in the GTM transmitter. The required rate for TXUSRCLK depends on the internal datapath width of the GTM_DUAL primitive and the TX line rate of the GTM transmitter. The following equation shows how to calculate the required rate for TXUSRCLK for all cases.

$$TXUSRCLK\ Rate = \frac{\text{Line Rate}}{Internal\ Datapath\ Width}$$

TXUSRCLK2 is the main synchronization clock for all signals into the TX side of the GTM transceiver. Most signals into the TX side of the GTM transceiver are sampled on the positive edge of TXUSRCLK2. TXUSRCLK2 and TXUSRCLK have a fixed-rate relationship based on the TX_DATA_WIDTH and TX_INT_DATA_WIDTH settings. The following table shows the relationship between TXUSRCLK2 and TXUSRCLK per TX_DATA_WIDTH and TX_INT_DATA_WIDTH values.

*Table 28:* **Relationship between TXUSRCLK2 and TXUSRCLK**

| Encoding | TX Data Width | TX Internal Datapath | TXUSRCLK2 Frequency |
|---|---|---|---|
| NRZ | 64 | 64 | $F_{TXUSRCLK2} = F_{TXUSRCLK}$ |
| NRZ | 128 | 64 | $F_{TXUSRCLK2} = F_{TXUSRCLK}/2$ |
| PAM4 | 80 | 80 | $F_{TXUSRCLK2} = F_{TXUSRCLK}$ |
| PAM4 | 160 | 80 | $F_{TXUSRCLK2} = F_{TXUSRCLK}/2$ |
| PAM4 | 128 | 128 | $F_{TXUSRCLK2} = F_{TXUSRCLK}$ |
| PAM4 | 256 | 128 | $F_{TXUSRCLK2} = F_{TXUSRCLK}/2$ |

These rules about the relationships between clocks must be observed for TXUSRCLK and TXUSRCLK2:

- TXUSRCLK and TXUSRCLK2 must be positive-edge aligned, with as little skew as possible between them. As a result, low-skew clock resources (BUFG_GTs) must be used to drive TXUSRCLK and TXUSRCLK2.
- Even though they might run at different frequencies, TXUSRCLK, TXUSRCLK2, and the transmitter reference clock must have the same oscillator as their source. Thus TXUSRCLK and TXUSRCLK2 must be multiplied or divided versions of the transmitter reference clock.

# Ports and Attributes

The following table defines the TX interface ports.

*Table 29:* **TX Interface Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| TXDATA[255:0] | In | TXUSRCLK2 | The bus for transmitting data. The width of this port is equal to the TX data width selection.<br>64: TXDATA[63:0].<br>80: TXDATA[79:0].<br>128: TXDATA[127:0].<br>160: TXDATA[159:0].<br>256: TXDATA[255:0]. |
| TXUSRCLK | In | Clock | This port is used to provide a clock for the internal TX PCS datapath. |

*Table 29:* **TX Interface Ports** *(cont'd)*

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| TXUSRCLK2 | In | Clock | This port is used to synchronize the interconnect logic with the TX interface. This clock must be positive-edge aligned to TXUSRCLK. |

The following table defines the TX interface attributes.

*Table 30:* **TX Interface Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_TX_PCS_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| TX_DATA_WIDTH | [2:0] | Sets the bit width of the TXDATA port. When FEC is enabled, TX_DATA_WIDTH must be set to 160:<br>`0x0`: 64-bit fabric mode.<br>`0x1`: 80-bit fabric mode.<br>`0x2`: 128 bit fabric mode.<br>`0x3`: 160-bit fabric mode.<br>`0x4`: 256-bit fabric mode. |
| TX_INT_DATA_WIDTH | [4:3] | Controls the width of the internal TX PCS datapath. 80-bit internal datapath must be used with 80- or 160-bit fabric width; 128-bit internal datapath must be used with 128- or 256-bit fabric width; 64-bit internal datapath must be used with 64- or 128-bit fabric width:<br>`0x0`: 64-bit internal datapath mode.<br>`0x1`: 80-bit internal datapath mode.<br>`0x2`: 128-bit internal datapath mode. |
| GEN_TXUSRCLK | [14] | Automatically generate TXUSRCLK from TXUSRCLK2. This is only applicable when the fabric datapath width is the same as the internal datapath width.<br>`0x0`: Disable automatic TXUSRCLK generation from TXUSRCLK2.<br>`0x1`: Enable automatic TXUSRCLK generation from TXUSRCLK2. |
| CH[0/1]_A_CH_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| TX_FABINT_USRCLK_FLOP | [0] | Determines if port signals are registered again in the TXUSRCLK domain after being registered in the TXUSRCLK2 domain. This attribute only applies if the TX internal datapath width is the same as the TX interface width, otherwise this attribute is ignored. Use the recommended value from the Wizard:<br>`0x0`: Bypass TXUSRCLK flip-flops.<br>`0x1`: Enable TXUSRCLK flip-flops. |

# Using TXPROGDIVCLK to Drive the TX Interface

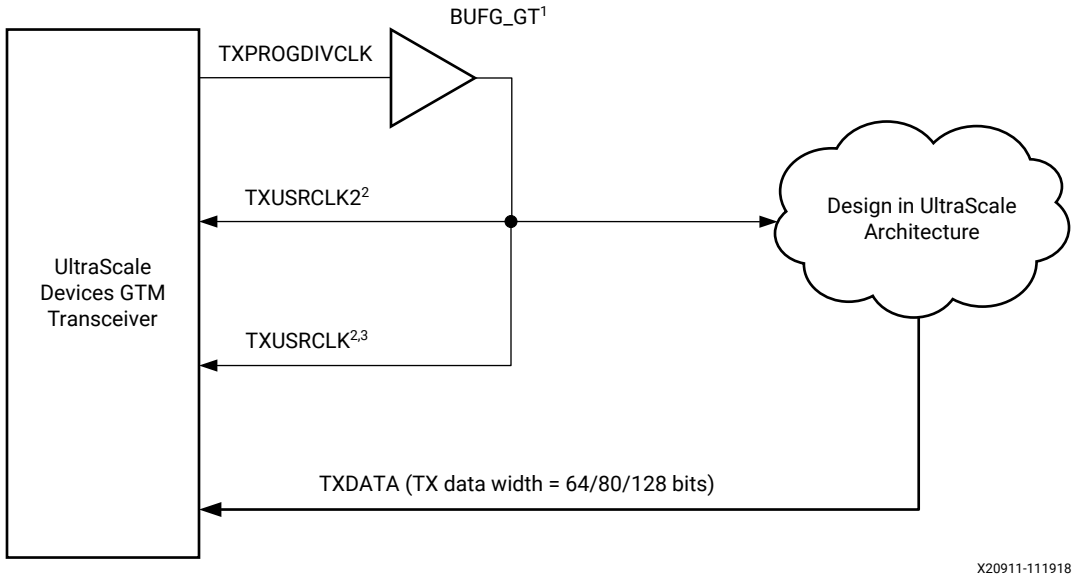
Depending on the TXUSRCLK and TXUSRCLK2 frequencies, there are different ways UltraScale architecture clock resources can be used to drive the parallel clock for the TX interface. Figure 27 through Figure 30 show different ways clock resources can be used to drive the parallel clocks for the TX interface.

Depending on the input reference clock frequency and the required line rate, a BUFG_GT with a properly configured divide setting is required. The UltraScale+ FPGAs GTM Transceivers Wizard creates a sample design based on different design requirements for most cases.

## *TXPROGDIVCLK Driving GTM Transceiver TX in 64-Bit, 80-Bit, or 128-Bit Mode*

In the following figure, TXPROGDIVCLK is used to drive TXUSRCLK and TXUSRCLK2 in 64-bit, 80-bit, or 128-bit mode in a single-lane configuration. In all cases, the frequency of TXUSRCLK2 is equal to TXUSRCLK.

*Figure 27:* **Single Lane—TXPROGDIVCLK Drives TXUSRCLK and TXUSRCLK2 (64-Bit, 80-Bit, or 128-Bit Mode)**

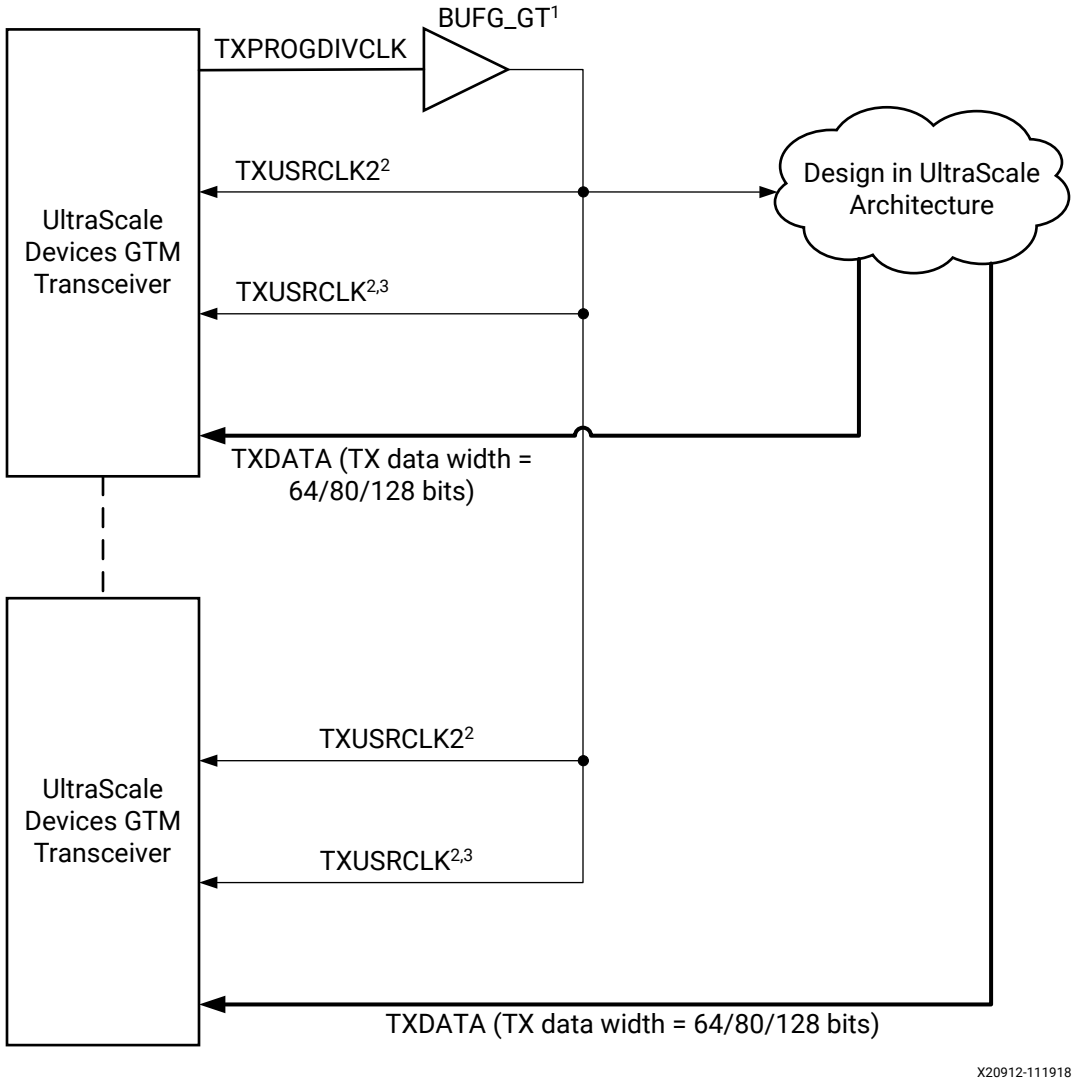



Notes relevant to the figure:

1. For details about placement constraints and restrictions on clocking resources (such as BUFG_GT and BUFG_GT_SYNC), refer to the *UltraScale Architecture Clocking Resources User Guide* (UG572).
2. $F_{TXUSRCLK2} = F_{TXUSRCLK}$.
3. TXUSRCLK can be tied to `1'b0` if GEN_TXUSRCLK = `1'b1`.

Similarly, the following figure shows the same settings in a multiple-lane configuration. In a multi-lane configuration, the middle-most GTM transceiver should be selected to be the source of TXPROGDIVCLK. For example, in a multi-lane configuration of six GTM transceivers consisting of three contiguous Duals, one of the middle GTM transceivers in the middle Dual should be selected as the source of TXPROGDIVCLK.

*Figure 28:* **Multiple Lanes—TXPROGDIVCLK Drives TXUSRCLK and TXUSRCLK2 (64-Bit, 80-Bit, or 128-Bit Mode)**




Notes relevant to the figure:

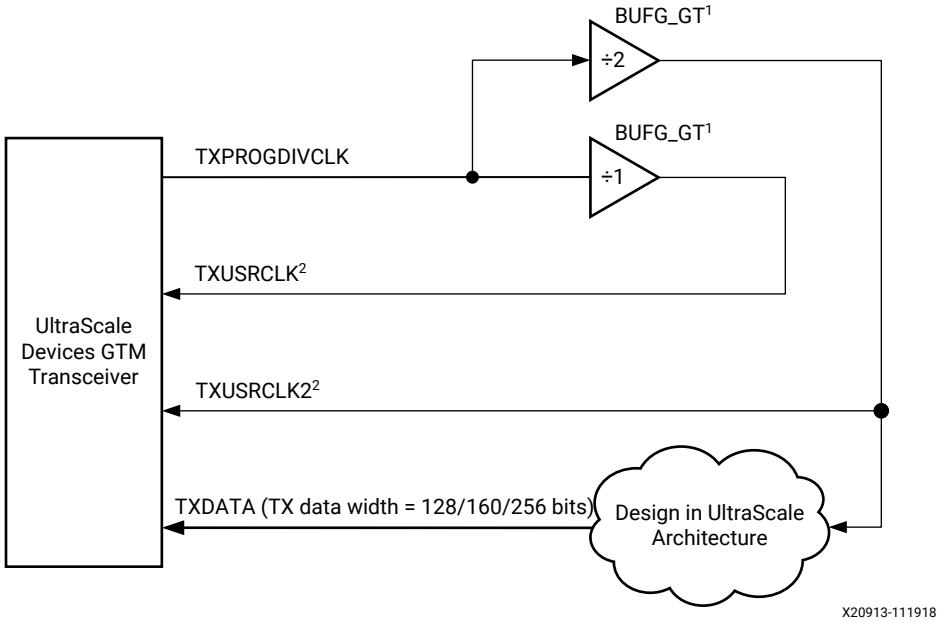
1. For details about placement constraints and restrictions on clocking resources (such as BUFG_GT and BUFG_GT_SYNC), refer to the *UltraScale Architecture Clocking Resources User Guide* (UG572).
2. $F_{TXUSRCLK2} = F_{TXUSRCLK}$.

3. TXUSRCLK can be tied to `1'b0` if GEN_TXUSRCLK = `1'b1`.

## *TXPROGDIVCLK Driving GTM Transceiver TX in 128-Bit, 160-Bit, or 256-Bit Mode*

In the following figure, TXPROGDIVCLK is used to drive TXUSRCLK and TXUSRCLK2 in 128-bit, 160-bit, or 256-bit mode in a single-lane configuration. In all cases, the frequency of TXUSRCLK2 is equal to half of the TXUSRCLK frequency.

*Figure 29:* **Single Lane—TXPROGDIVCLK Drives TXUSRCLK and TXUSRCLK2 (128-Bit, 160-Bit, or 256-Bit Mode)**

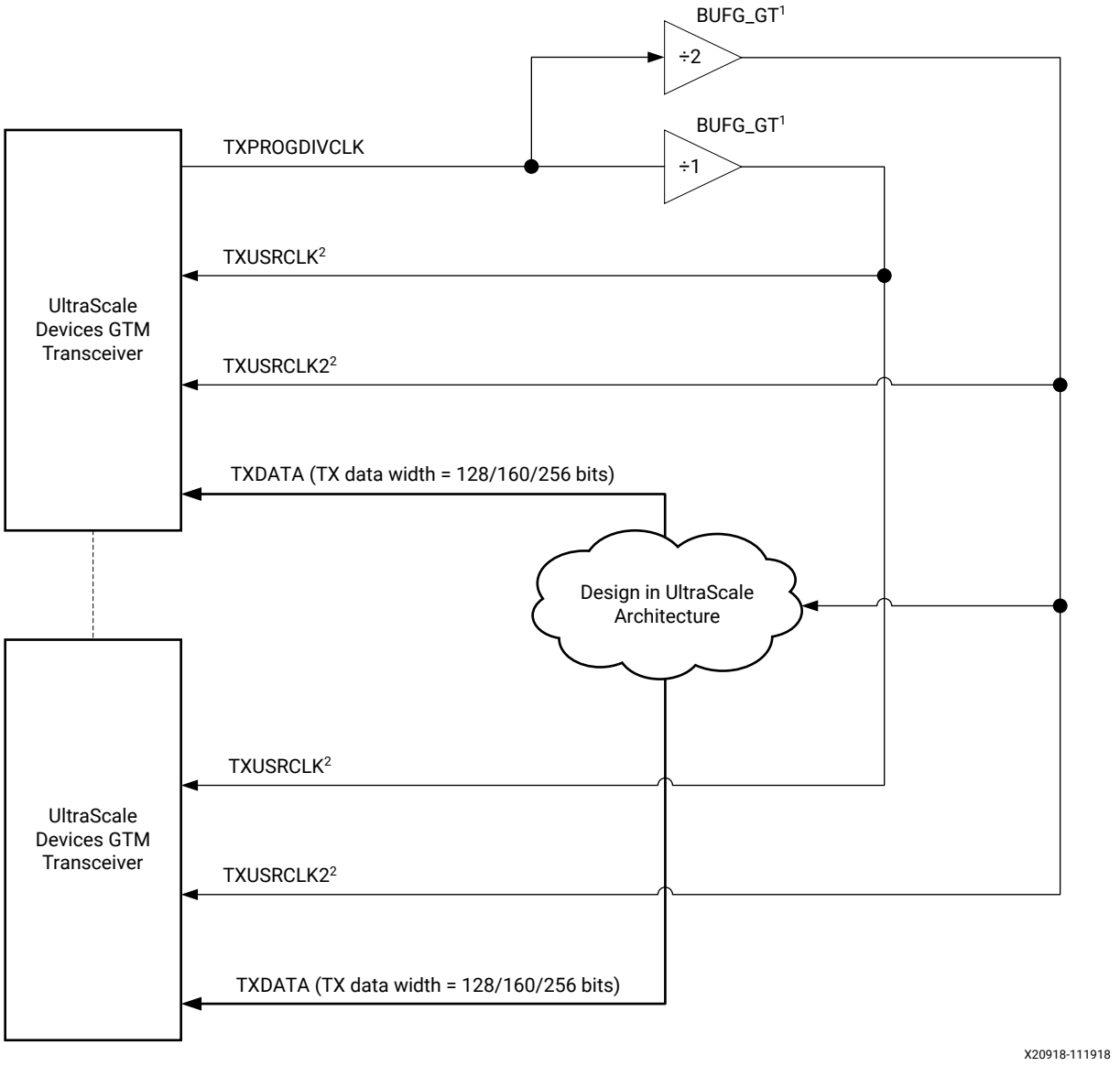



Notes relevant to the figure:

1. For details about placement constraints and restrictions on clocking resources (such as BUFG_GT and BUFG_GT_SYNC), refer to the *UltraScale Architecture Clocking Resources User Guide* (UG572).
2. $F_{TXUSRCLK2} = F_{TXUSRCLK}/2$.

Similarly, the following figure shows the same settings in a multiple-lane configuration. In a multi-lane configuration, the middle-most GTM transceiver should be selected to be the source of TXPROGDIVCLK. For example, in a multi-lane configuration of six GTM transceivers consisting of three contiguous Duals, one of the middle GTM transceivers in the middle Dual should be selected as the source of TXPROGDIVCLK.

*Figure 30:* **Multiple Lanes—TXPROGDIVCLK Drives TXUSRCLK and TXUSRCLK2 (128-Bit, 160-Bit, or 256-Bit Mode)**




Notes relevant to the figure:

1. For details about placement constraints and restrictions on clocking resources (BUFG_GT, BUFG_GT_SYNC, etc.), refer to the *UltraScale Architecture Clocking Resources User Guide* (UG572).
2. $F_{TXUSRCLK2} = F_{TXUSRCLK}/2$.

# TX FEC

The Integrated KP4 Reed-Solomon Forward Error Correction (RS-FEC) provides a robust multi-bit error detection/correction algorithm that protects up to 2 x 58 Gb/s or 1 x 116 Gb/s electrical and optical links. This section describes the operation of the Integrated KP4 RS-FEC within the UltraScale+™ device GTM transceivers.

KP4 FEC is based on the RS(544,514) code, which encodes message blocks of 5140 bits to produce codewords of 5440 bits. For a detailed description of the RS-FEC sublayer in Ethernet, including the definition of the KP4 FEC code, refer to clause 91 of the IEEE Standard for Ethernet (IEEE Std 802.3-2015). The same FEC code is used in other standards such as OTN FlexO and Interlaken.

The Integrated KP4 RS-FEC for each GTM dual is composed of two logical slices for each channel. These can operate as two independent RS-FEC processing units at up to 58 Gb/s each, or as one unified unit at up to 116 Gb/s. When operating at up to 116 Gb/s, data is transmitted and received over four virtual FEC lanes as described in IEEE 802.3-2015 clause 91. When operating as 2 x 58 Gb/s, data can be transmitted and received over two virtual FEC lanes per 58 Gb/s channel, or as a raw data stream (one virtual lane) with optional PN scrambling for backplane operations. The general principle of operation of the FEC is the same whichever mode is chosen.

When RS-FEC is enabled in the transmit direction, data to be FEC-encoded and transmitted is provided from the fabric to the input of the GTM transceiver. The pre-FEC data must be pre-formatted to contain zero padding regions where the parity will be inserted. The Integrated KP4 RS-FEC performs RS encoding to fill in the parity space, and (except in raw mode) also performs symbol distribution operations according to the 802.3bj clause 91 specification. The encoded output data from the Integrated KP4 RS-FEC is then presented to the GTM PCS for transmission.

The Integrated KP4 RS-FEC does not natively perform transcoding, alignment marker removal, alignment marker mapping, or alignment marker insertion operations in the transmit directions. To support protocols such as 100G Ethernet (IEEE 802.3 clause 91) and 50G Ethernet (IEEE 802.3 clause 134) which require 257b transcoding and alignment marker processing, the GTM Wizard IP can optionally include soft logic blocks for these functions.

## Ports and Attributes

The following table shows the TX FEC-related ports for the GTM dual.

*Table 31:* **TX FEC Ports**

| Ports | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_TXFECRESET | In | Async | Component reset port for TX FEC. |

*Table 31:* **TX FEC Ports** *(cont'd)*

| Ports | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_TXDATA[159:0] | In | CH[0/1]_TXUSRCLK2 | Input TX data, must use 160-bit interface when FEC is enabled. |
| CH[0/1]_TXDATASTART | In | CH[0/1]_TXUSRCLK2 | Start of codeword. |

The transmit portion of the Integrated KP4 RS-FEC operates internally in the CH0_TXUSRCLK and CH1_TXUSRCLK domains. Data input on CH0_TXDATA is clocked on the rising edge of CH0_TXUSRCLK2, and data input on CH1_TXDATA is clocked on the rising edge of CH1_TXUSRCLK2, just as when the FEC is not enabled.

When configured in 100G mode (combined slice 0 and slice 1 operation), all data from both channels must be driven by the CH0_TXUSRCLK2 clock. The CH1_TXUSRCLK and CH1_TXUSRCLK2 inputs can be tied to ground.

The following table shows the TX FEC-related attributes for the GTM dual.

*Table 32:* **TX FEC Attributes**

| Attribute | Type | Description |
|---|---|---|
| FEC_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| FEC_TX0_MODE | [3:0] | Operation mode for FEC TX slice 0:<br>`4'b0000`: FEC is disabled for this channel.<br>`4'b0001`: 50G KP4 FEC, 50GAUI-1 format.<br>`4'b0010`: 100G KP4 FEC, 100GAUI-2 format.<br>`4'b0101`: 50G raw KP4 FEC without scrambling.<br>`4'b1101`: 50G raw KP4 FEC with scrambling.<br>Others: Invalid. |
| FEC_TX1_MODE | [7:4] | Operation mode for FEC TX slice 1:<br>`4'b0000`: FEC is disabled for this channel.<br>`4'b0001`: 50G KP4 FEC, 50GAUI-1 format.<br>`4'b0010`: 100G KP4 FEC, 100GAUI-2 format.<br>`4'b0101`: 50G raw KP4 FEC without scrambling.<br>`4'b1101`: 50G raw KP4 FEC with scrambling.<br>Others: Invalid. |

# Usage Model

When FEC is enabled in 50G mode, bit 0 of each TXDATA bus must be the first bit to be transmitted in time, and bit 159 must be the last bit to be transmitted in time. The TXDATASTART signal must be driven High whenever the TXDATA for the associated channel contains the first 160 bits of a codeword. Input codewords must always be aligned so that bit 0 of a codeword is on bit 0 of the TXDATA bus.

When FEC is enabled in 100G mode, bit 0 of CH0_TXDATA must be the first bit to be transmitted in time, and bit 159 of CH1_TXDATA must be the last bit to be transmitted in time. The CH0_TXDATASTART signal must be driven High whenever the TXDATA buses contain the first 320 bits of a codeword. Input codewords must always be aligned so that bit 0 of a codeword is on bit 0 of CH0_TXDATA. CH1_TXDATASTART is ignored in 100G mode.

In all modes, codewords are 5440 bits in length. The final 300 bits of data is reserved for the insertion of FEC parity. At the input to the FEC, bits 5140 to 5439 of the input data must be set to 0.

## *50G Ethernet*

Up to two channels of 50G Ethernet with KP4 FEC can be implemented as per *IEEE Draft Standard for Ethernet Amendment: Media Access Control Parameters for 50 Gb/s and Physical Layers and Management Parameters for 50 Gb/s, 100 Gb/s, and 200 Gb/s Operation* (IEEE Std 802.3cd Clause 134). Transcoding should be enabled in the GTM Wizard IP for this mode. The nominal pre-FEC PCS data rate is 51.5625 Gb/s, and the nominal post-FEC line rate is 53.125 Gb/s.

## *100G Ethernet*

One channel of 100G Ethernet with KP4 FEC can be implemented as per IEEE Std 802.3-2015 Clause 91. Transcoding should be enabled in the GTM Wizard IP for this mode. The nominal aggregate pre-FEC PCS data rate is 103.125 Gb/s and the nominal aggregate post-FEC line rate is 106.25 Gb/s.

## *100G OTN FlexO*

One channel of 100G OTN FlexO with a KP4 FEC can be implemented as per ITU-T G.709.1, *Flexible OTN Short-Reach Interface*. Transcoding should be disabled in the GTM Wizard IP for this mode. The nominal aggregate post-FEC line rate is 111.81 Gb/s.

## *100G Interlaken*

One channel of 100G Interlaken with KP4 FEC can be implemented as per the *Interlaken Reed-Solomon Forward Error Correction Extension Protocol Definition*. Transcoding should be disabled in the GTM Wizard IP for this mode. Line rates up to 58 Gb/s are possible.

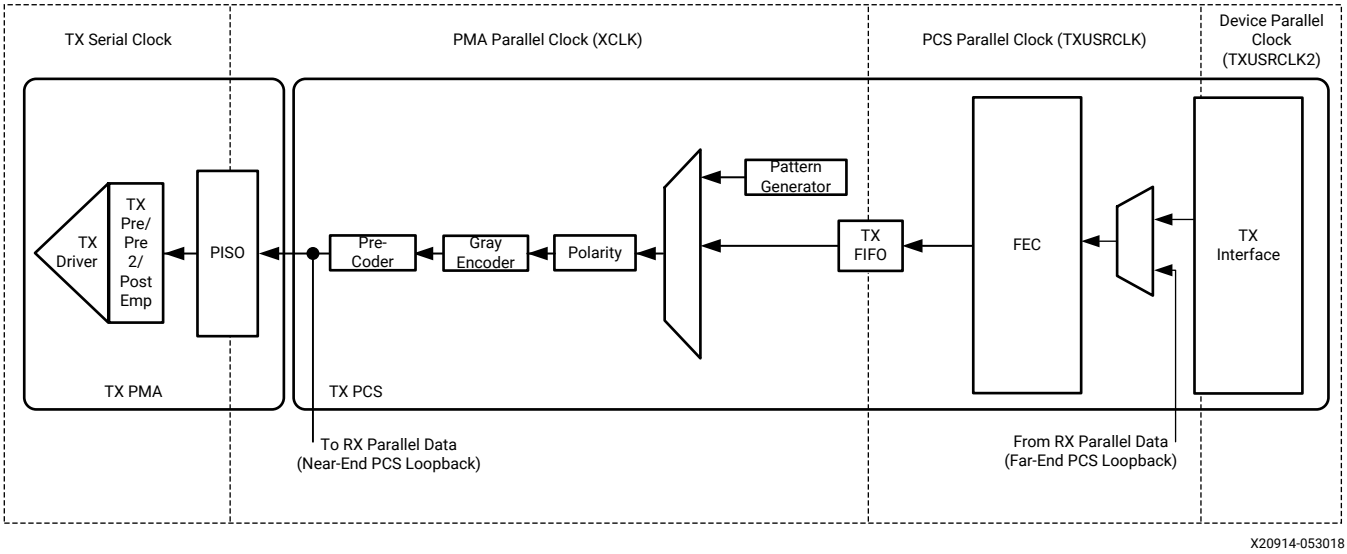
### *Proprietary Backplane Protocols with FEC up to 58 Gb/s*

FEC can be enabled in 50G raw mode (with or without scrambling).

# TX Buffer

The GTM transceiver TX datapath has two internal parallel clock domains used in the PCS: the interface with PMA parallel clock domian (XCLK), and the PCS internal clock domain (TXUSRCLK). To transmit data, the TX buffer provides data width conversion between these clock domains when necessary, depending on the operating data width and encoding mode. The following figure shows the TX datapath clock domains.

*Figure 31:* **TX Clock Domains**




The GTM transmitter includes a TX FIFO to support data width conversion when data crosses from TXUSRCLK to XCLK domain, and the table below shows the possible scenarios. The buffer does not tolerate ppm differences, and only provides phase compensation between the two clocks. The TX buffer inside the GTM transceiver must always be used. Buffer bypass is not allowed.

*Table 33:* **TX FIFO Data Width Conversion Scenarios**

| PCS Parallel Clock (TXUSRCLK) Domain Data Width | PMA Parallel Clock (XCLK) Domain Data Width | FEC Support |
|---|---|---|
| 64-bit | 64-bit | No |
| 80-bit | 128-bit | Yes |
| 128-bit | 128-bit | No |

# Ports and Attributes

The following table defines the TX buffer ports.

*Table 34:* **TX Buffer Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_TXBUFSTATUS[1:0] | Out | TXUSRCLK | TX buffer status:<br>Bit[1]: FIFO overflow status. A value of 1 indicates FIFO overflow.<br>Bit[0]: FIFO underflow status. A value of 1 indicates FIFO underflow. |

The following table defines the TX buffer attributes.

*Table 35:* **TX Buffer Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_TX_PCS_CFG0 | 16-bit | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_TXFIFO_UNDERFLOW | 1-bit | A value of 1 indicates FIFO underflow.<br>For channel 0, bit[8] of DRP address `0x489`<br>For channel 1, bit[8] of DRP address `0x689`<br>***Note*:** This is a read-only attribute. |
| CH[0/1]_TXFIFO_OVERFLOW | 1-bit | A value of 1 indicates FIFO overflow.<br>For channel 0, bit[9] of DRP address `0x489`<br>For channel 1, bit[9] of DRP address `0x689`<br>***Note*:** This is a read-only attribute. |

# Using the TX Buffer

Reset the TX buffer whenever CH[0/1]_TXBUFSTATUS indicates an overflow or underflow condition. The TX buffer can be reset using CH[0/1]_GTTXRESET and CH[0/1]_TXPCSRESET (see TX Initialization and Reset).

# TX Pattern Generator

Pseudo-random it sequences (PRBS) are commonly used to test the signal integrity of high-speed links. These sequences appear random but have specific properties that can be used to measure the quality of a link. The GTM transceiver pattern generator block can generate several industry-standard PRBS patterns listed in the following table.

*Table 36:* **Supported PRBS Patterns**

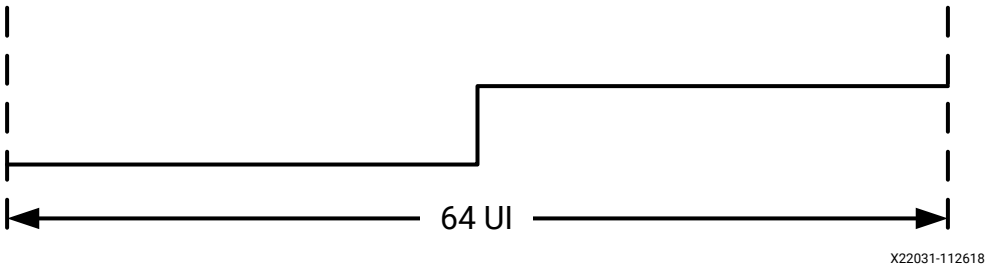
| Name | Polynomial | Length of Sequence | Description |
|---|---|---|---|
| PRBS-7 | $1 + x^6 + x^7$ | $2^7 - 1$ bits | Used to test channels with 8B/10B. |
| PRBS-9 | $1 + x^5 + x^9$ | $2^9 - 1$ bits | ITU-T Recommendation O.150, Section 5.1. PRBS-9 is one of the recommended test patterns for SFP+. |
| PRBS-13 | $1 + x + x^2 + x^{12} + x^{13}$ | $2^9 - 1$ bits | IEEE Std P802.3bs 120.5.11.2.1 test requires PRBS13Q test pattern. OIF2014.230 CEI-56G-VSR-PAM4 specification requires QPRBS13-CEI test pattern. |
| PRBS-15 | $1 + x^{14} + x^{15}$ | $2^{15} - 1$ bits | ITU-T Recommendation O.150, Section 5.3. PRBS-15 is often used for jitter measurement because it is the longest pattern the Agilent DCA-J sampling scope can handle. |
| PRBS-23 | $1 + x^{18} + x^{23}$ | $2^{23} - 1$ bits | ITU-T Recommendation O.150, Section 5.6. PRBS-23 is often used for non-8B/10B encoding schemes. It is one of the recommended test patterns in the SONET specification. |
| PRBS-31 | $1 + x^{28} + x^{31}$ | $2^{31} - 1$ bits | ITU-T Recommendation O.150, Section 5.8. PRBS-31 is often used for non-8B/10B encoding schemes. It is a recommended PRBS test pattern for 10 Gigabit Ethernet. See IEEE Std 802.3ae-2002. IEEE Std P802.3bs120.5.11.2.2 test requires PRBS31Q test pattern. |

**IMPORTANT!** *For PAM4 modulation, QPRBS/PRBSQ patterns are supported by sending a conventional PRBS pattern with PAM4 and Gray Coding based on the OIF2014.230 CEI-56G-VSR-PAM4 specification and IEEE Std P802.3bs.*

In addition to PRBS patterns, the GTM transceiver supports a 64 UI square wave test pattern as shown in the following figure, an alternating `1'b0` and `1'b1` (NRZ clock) test pattern. Clocking patterns are usually used to check PLL random jitter often done with a spectrum analyzer.

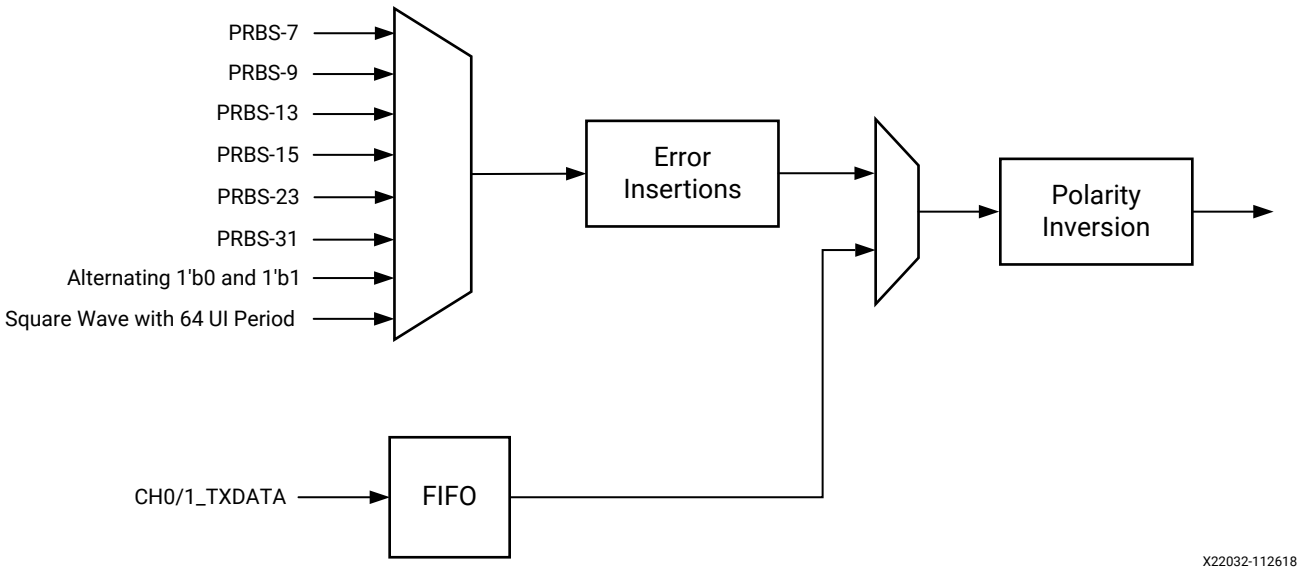
***Note*:** For PAM4 modulation, an alternating `1'b0` and `1'b1` test pattern will not be a square wave due to the amplitude modulation mapping.

*Figure 32:* **64 UI Square Wave**




The error insertion function is also supported to verify link connection for jitter tolerance tests. When an inverted PRBS pattern is necessary, the CH[0/1]_TXPOLARITY signal is used to control polarity.

*Figure 33:* **TX Pattern Generator Block**




# Ports and Attributes

The following table defines the pattern generator ports.

*Table 37:* **Pattern Generator Ports**

| Port Name | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_TXPRBSPTN[3:0] | In | CH[0/1]_TXUSRCLK2 | Transmitter pattern generator control.<br>`4'b0000`: Standard operation mode (test pattern generation is off)<br>`4'b0001`: PRBS-7<br>`4'b0010`: PRBS-9<br>`4'b0011`: PRBS-15<br>`4'b0100`: PRBS-23<br>`4'b0101`: PRBS-31<br>`4'b0110`: PRBS-13<br>`4'b1000`: Reserved<br>`4'b1001`: Alternating `1b'0` and `1'b1`<br>`4'b1010`: Square wave with 64 UI period |
| CH[0/1]_TXPRBSINERR | In | CH[0/1]_TXUSRCLK2 | When this port is driven High, a single error is forced in the PRBS transmitter for every CH[0/1]_TXUSRCLK2 clock cycle that the port is asserted.<br>When CH[0/1]_TXPRBSPTN is set to `4'b0000`, this port does not affect CH[0/1]_TXDATA. |
| CH[0/1]_TXQPRBSEN | In | CH[0/1]_TXUSRCLK2 | Reserved. This port must always be set to `1'b0`. |

The following table defines the pattern generator attribute.

*Table 38:* **Pattern Generator Attribute**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_TX_PCS_CFG1 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| RXPRBSERR_LOOPBACK | [7] | Setting this attribute to `1'b1` causes the CH[0/1]_RXPRBSERR bit to be internally looped back to CH[0/1]_TXPRBSINERR of the same GTM transceiver. This allows synchronous and asynchronous jitter tolerance testing without worrying about data clock domain crossing. Setting this attribute to `1'b0` causes CH[0/1]_TXPRBSINERR to be forced onto the TX PRBS. |

# Using TX Pattern Generator

The GTM TX pattern generator works for all supported data widths. However, 80-bit or 160-bit TX fabric data width in PAM4 mode requires additional steps. Other data widths do not require any additional steps.

### Enable TX Pattern Generator for 80-bit or 160-bit Data Width

1. Using the DRP interface, write the following values to CH[0/1]_TX_PCS_CFG0[4:0] (address 0x083 for CH0, 0x283 for CH1):
   a. CH[0/1]_TX_PCS_CFG0[4:0] = 0x12 for 80-bit data width mode.
   b. CH[0/1]_TX_PCS_CFG0[4:0] = 0x14 for 160-bit data width mode.
2. Enable the PRBS generator by setting CH[0/1]_TXPRBSPTN to the required value for the desired pattern.
3. TX PRBS pattern generator is enabled.

***Note*:** Toggling CH[0/1]_TXPRBSINSERR for one CH[0/1]_TXUSRCLK2 cycle might inject more than one error into the pattern generator in 80/160 bit mode.

### Disable TX Pattern Generator for 80-bit or 160-bit Data Width

1. Disable the PRBS generator by setting CH[0/1]_TXPRBSPTN[3:0] to `4'b0000`.
2. Using the DRP interface, write the following values to CH[0/1]_TX_PCS_CFG0[4:0] (address 0x083 for CH0, 0x283 for CH1):
   a. CH[0/1]_TX_PCS_CFG0[4:0] = 0x09 for 80-bit data width mode.
   b. CH[0/1]_TX_PCS_CFG0[4:0] = 0x0B for 160-bit data width mode.
3. Set CH[0/1]_TXPMARESETMASK = 0x0.
4. Toggle CH[0/1]_GTTXRESET High and Low.
5. Wait for CH[0/1]_TXRESETDONE to toggle High.
6. Set CH[0/1]_TXPMARESETMASK = 0x3.

TX PRBS pattern generator is disabled and the TX is driven based on the CH[0/1]_TXDATA input.

# TX Polarity Control

If TXP and TXN differential traces are accidentally swapped on the PCB, the differential data transmitted by the GTM transceiver TX is reversed. One solution is to invert the parallel data before serialization and transmission to offset the reversed polarity on the differential pair. The TX polarity control can be accessed through the CH0_TXPOLARITY and CH1_TXPOLARITY input from the interconnect logic interface. The TX polarity control is driven High to invert the polarity of outgoing data.

## Ports and Attributes

The following table defines the ports required for TX polarity control.

*Table 39:* **TX Polarity Control Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH0_TXPOLARITY[1] | In | CH0_TXUSRCLK2 | The CH0_TXPOLARITY port is used to invert the polarity of the outgoing data for channel 0:<br>0: Not inverted. TXP is positive, and TXN is negative.<br>1: Inverted. TXP is negative, and TXN is positive. |
| CH1_TXPOLARITY[1] | In | CH1_TXUSRCLK2 | The CH1_TXPOLARITY port is used to invert the polarity of the outgoing data for channel 1:<br>0: Not inverted. TXP is positive, and TXN is negative.<br>1: Inverted. TXP is negative, and TXN is positive. |

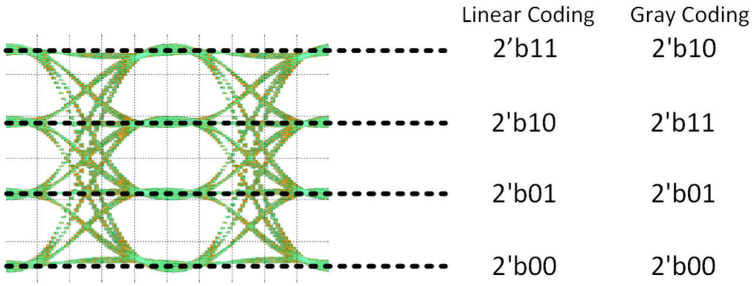
**Notes:**

1. CH[0/1]_TXPOLARITY can be tied High if the polarity of TXP and TXN needs to be reversed.

# TX Gray Encoder

GTM transmitters in UltraScale+ devices support two types of binary encoding options: linear coding and Gray coding. By using Gray coding, only one bit error per symbol is made for incorrect decisions, thus reducing the bit-error rate by more than 33%. The following figure illustrates the differences between linear coding and Gray coding.

*Figure 34:* **Transmitted PAM4 Signal Bit Encoding**



## Ports and Attributes

The following table defines the attributes required for TX Gray encoder control.

*Table 40:* **Gray Encoder Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_TX_PCS_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| TX_GRAY_ENDIAN | [13] | In PAM4 mode, this attribute controls transmitted endianness. In NRZ mode, the default Wizard value must be used.<br>`1'b0`: Big endian.<br>`1'b1`: Little endian. |
| TX_GRAY_BYP_EN | [12] | In PAM4 mode, this attribute enables Gray encoding. In NRZ mode, the default Wizard value must be used.<br>`1'b0`: Enables Gray encoding.<br>`1'b1`: Disables Gray encoding. |

**IMPORTANT!** *In PAM4 mode, if Gray encoder is enabled for the transmitter, the receiver Gray decoder should also be enabled for proper data recovery.*

# TX Pre-Coder

GTM transmitters in UltraScale+ devices support pre-coding. Pre-coding can be used to reduce receiver decision feedback equalization (DFE) error propagation by reducing 1-tap burst error runs into two errors for every error event.

## Ports and Attributes

The following table defines the attributes required for TX pre-coder control.

*Table 41:* **Pre-Coder Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_TX_PCS_CFG0 | 16-bit | Reserved. |

*Table 41:* **Pre-Coder Attributes** *(cont'd)*

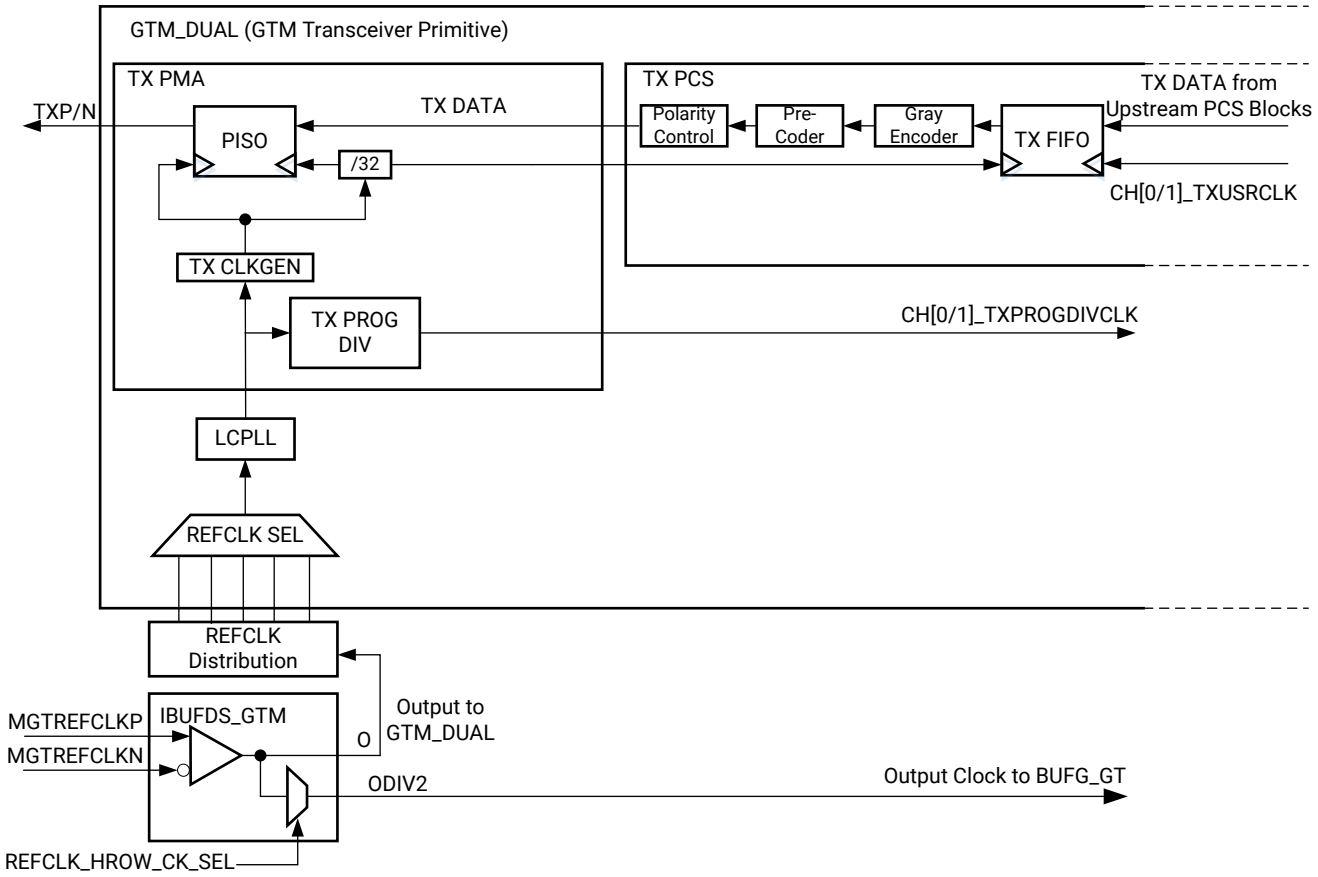
| Attribute | Type | Description |
|---|---|---|
| **Bit Name** | **Address** | **Description** |
| TX_PRECODE_ENDIAN | [11] | In PAM4 mode, this attribute controls pre-coder transmitted endianness. In NRZ mode, the default Wizard value must be used.<br>`1'b0`: Big endian.<br>`1'b1`: Little endian. |
| TX_PRECODE_BYP_EN | [10] | In PAM4 mode, this attribute enables pre-coding. In NRZ mode, the default Wizard value must be used.<br>`1'b0`: Enables pre-code.<br>`1'b1`: Disables pre-code. |

**IMPORTANT!** *In PAM4 mode, if pre-coder is enabled for the transmitter, the receiver pre-coder should also be enabled for proper data recovery.*

# TX Fabric Clock Output Control

The TX Clock Divider Control block has two main components: serial clock divider control, and parallel clock divider and selector control. The clock divider and selector details are illustrated in the following figure.

*Figure 35:* **TX Serial and Parallel Clock Divider**




Notes relevant to the figure:

1. CH[0/1]_TXPROGDIVCLK is used as the source of the interconnect logic clock via BUFG_GT.
2. There is only one LCPLL in the GTM_DUAL primitive, which is shared between the TX/RX.

# TX Programmable Divider

The TX programmable divider shown in Figure 35 uses the LCPLL output clock to generate a parallel output clock. By using the transceiver LCPLL, TX programmable divider, and BUFG_GT, CH[0/1]_TXPROGDIVCLK should be used as a clock source for the interconnect logic.

The following tables show the programmable divider ports and attributes, respectively.

*Table 42:* **TX Programmable Divider Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_TXPROGDIVRESET | In | Async | This active-High port resets the dividers as well as the TXPRGDIVRESETDONE indicator. A reset must be performed whenever the input clock source is interrupted. |
| CH[0/1]_TXPRGDIVRESETDONE | Out | Async | When the input clock is stable and reset is performed, this active-High signal indicates the reset is completed and the output clock is stable. |
| CH[0/1]_TXPROGDIVCLK | Out | Clock | TXPROGDIVCLK is the parallel clock output from the TX programmable divider. This clock is the recommended output to the interconnect logic through BUFG_GT. |

*Table 43:* **TX Programmable Divider Attribute**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_TX_DRV_CFG4 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| TX_PROGDIV_SEL_FULLRATE | [15] | This attribute is used during the TX programmable divider ratio selection. Set to `1'b1` to obtain the full rate of the divided clock. Set to `1'b0` to obtain the half rate of the divided clock. |

*Table 43:* **TX Programmable Divider Attribute** *(cont'd)*

| Attribute | Type | Description |
|---|---|---|
| TX_PROGDIV_FBKDIV | [11:6] | This attribute is the main TX programmable divider selector. When the following settings are set:<br>TX_PROGDIV_SEL_DIV66 = `1'b1`<br>TX_PROGDIV_PDBV_DIV5 = `1'b0`<br>TX_PROGDIV_SEL_FULLRATE = `1'b1` (or `1'b0`)<br>Valid TX programmable divider ratios are:<br>`6'b011000`: 4 (8)<br>`6'b111000`: 5 (10)<br>`6'b000000`: 8 (16)<br>`6'b100000`: 10 (20)<br>`6'b000001`: 12 (24)<br>`6'b100001`: 15 (30)<br>`6'b000010`: 16 (32)<br>`6'b100010`: 20 (40)<br>`6'b000101`: 24 (48)<br>`6'b100011`: 25 (50)<br>`6'b100101`: 30 (60)<br>`6'b000110`: 32 (64)<br>`6'b100110`: 40 (80)<br>`6'b001101`: 48 (96)<br>`6'b100111`: 50 (100)<br>`6'b101101`: 60 (120)<br>`6'b001110`: 64 (128)<br>`6'b001111`: 80 (160)<br>`6'b101111`: 100 (200)<br>When the following settings are set:<br>TX_PROGDIV_SEL_DIV66 = `1'b0`<br>TX_PROGDIV_PDBV_DIV5 = `1'b1`<br>TX_PROGDIV_SEL_FULLRATE = `1'b1` (or `1'b0`)<br>Valid TX programmable divider ratios are:<br>`6'b011000`: 16.5 (33)<br>`6'b000000`: 33 (66)<br>`6'b000010`: 66 (132) |
| TX_PROGDIV_SEL_DIV66 | [3] | This attribute is used during the TX programmable divider ratio selection.<br>• The attribute must be set to `1'b1` when the desired divider value is either 16.5, 33, 66, or 132.<br>• For all other divider values, this should be set to `1'b0`. |

*Table 43:* **TX Programmable Divider Attribute** *(cont'd)*

| Attribute | Type | Description |
|---|---|---|
| TX_PROGDIV_PDBV_DIV5 | [2] | This attribute is used during the TX programmable divider ratio selection.<br>• The attribute must be set to `1'b0` when the desired divider value is either 16.5, 33, 66, or 132.<br>• For all other divider values, this should be set to `1'b1`. |

## Ports and Attributes

The following table defines the ports required for TX fabric clock output control.

*Table 44:* **TX Fabric Clock Output Control Ports**

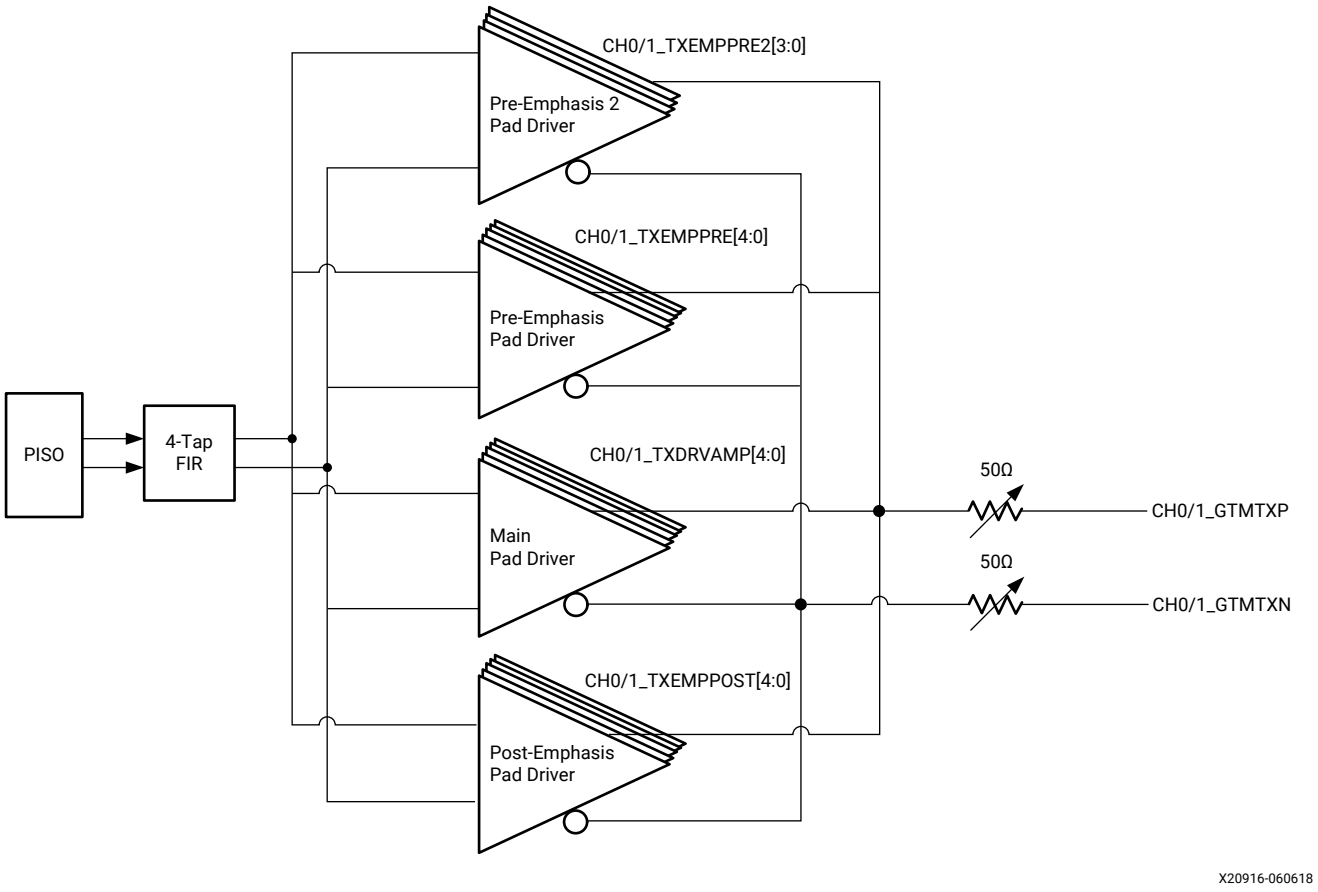
| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_TXOUTCLKSEL[2:0] | In | Async | This port must be set to `3'b000`. |
| CH[0/1]_TXOUTCLK | Out | Clock | Reserved. |
| CH[0/1]_TXPROGDIVCLK | Out | Clock | TXPROGDIVCLK is the parallel output clock from the TX programmable divider. This clock is the recommended output to the interconnect logic through BUFG_GT. |

# TX Configurable Driver

The GTM transceiver TX driver is a high-speed voltage-mode differential output buffer. To maximize signal integrity, it includes these features:

- Differential voltage control
- Two pre-cursor, and one post-cursor transmit pre-emphasis
- Two modulation schemes: NRZ and PAM4
- Calibrated termination resistors

*Figure 36:* **TX Configurable Driver Block Diagram**




# Ports and Attributes

The following table defines the TX configurable driver ports.

*Table 45:* **TX Configurable Driver Ports**

| Port | Dir | Clock Domain | Description | |
|---|---|---|---|---|
| CH[0/1]_TXDRVAMP[4:0] | Input | Async | Driver swing control. The default is user specified. All listed values are in $mV_{PPD}$. | |
| | | | **[4:0]** | **$mV_{PPD}$** |
| | | | 5'b00000 | 222 |
| | | | 5'b00001 | 245 |
| | | | 5'b00010 | 268 |
| | | | 5'b00011 | 292 |
| | | | 5'b00100 | 316 |
| | | | 5'b00101 | 340 |
| | | | 5'b00110 | 364 |

*Table 45:* **TX Configurable Driver Ports** *(cont'd)*

| Port | Dir | Clock Domain | Description | |
|---|---|---|---|---|
| | | | 5'b00111 | 388 |
| | | | 5'b01000 | 411 |
| | | | 5'b01001 | 435 |
| | | | 5'b01010 | 459 |
| | | | 5'b01011 | 483 |
| | | | 5'b01100 | 506 |
| | | | 5'b01101 | 531 |
| | | | 5'b01110 | 554 |
| | | | 5'b01111 | 578 |
| | | | 5'b10000 | 601 |
| | | | 5'b10001 | 627 |
| | | | 5'b10010 | 651 |
| | | | 5'b10011 | 675 |
| | | | 5'b10100 | 698 |
| | | | 5'b10101 | 722 |
| | | | 5'b10110 | 736 |
| | | | 5'b10111 | 761 |
| | | | 5'b11000 | 784 |
| | | | 5'b11001 | 808 |
| | | | 5'b11010 | 833 |
| | | | 5'b11011 | 858 |
| | | | 5'b11100 | 880 |
| | | | 5'b11101 | 903 |
| | | | 5'b11110 | 927 |
| | | | 5'b11111 | 949 |
| | | | **Notes:**<br>1. The peak-to-peak differential voltage is defined when CH[0/1]_TXEMPPOST = 5'b00000, CH[0/1]_TXEMPPRE = 5'b00000, and CH[0/1]_TXEMPPRE2 = 4'b0000.<br>2. For UltraScale+ FPGAs, the output swing described above is obtained using settings from the Wizard design, and the recommended values from the Wizard should not be changed. | |
| CH[0/1]_TXINHIBIT | Input | TXUSRCLK2 | When High, this signal blocks transmission of CH[0/1]_TXDATA and forces CH[0/1]_GTMTXP to NRZ logic 0 and CH[0/1]_GTMTXN to NRZ logic 1. This applies for both NRZ and PAM4 modes. | |

*Table 45:* **TX Configurable Driver Ports** *(cont'd)*

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_TXEMPMAIN[5:0] | Input | | Allows the main cursor coefficients to be directly set if the CH[0/1]_TX_DRV_CFG0[0] attribute is set to `1'b1`.<br>CH[0/1]_TXDRVAMP should be used together with CH[0/1]_TXEMPMAIN to achieve the desired TX output swing. For 10G-KR operation:<br>1. CH[0/1]_TXCTLFIRDAT [5:4] = `2'b01`<br>FIR restrictions and boundaries:<br>1. Pre-cursors allowed range is 0 to 15 coefficient units<br>a. Set pre-cursor using CH[0/1]_TXEMPPRE<br>2. Post-cursor allowed range is 0 to 33 coefficient units<br>a. If post-cursor coefficient units < 16<br>i. Set CH[0/1]_TXEMPPRE2 = 0 coefficient units<br>ii. Set post-cursor using CH[0/1]_TXEMPPOST<br>b. If post-cursor is 16 ≤ coefficient units ≤ 23<br>i. Set CH[0/1]_TXEMPPRE2 = 4 coefficient units<br>ii. Set CH[0/1]_TXEMPPOST = post-cursor coefficient units – 4<br>c. If post-cursor is 24 ≤ coefficient units ≤ 33<br>i. Set CH[0/1]_TXEMPPRE2 = 8 coefficient units<br>ii. Set CH[0/1]_TXEMPPOST = post-cursor coefficient units – 8<br>3. Coefficient units for pre-cursor, post-cursor, and main-cursor need to satisfy the following condition:<br>a. CH[0/1]_TXEMPMAIN coefficient units ≥ CH[0/1]_TXEMPPRE coefficient units + CH[0/1]_TXEMPPOST coefficient units ≤ 39 |

*Table 45:* **TX Configurable Driver Ports** *(cont'd)*

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_TXEMPPRE[4:0] | Input | Async | Transmitter pre-cursor TX pre-emphasis control. The default is user specified. All listed values (dB) are typical |

| [4:0] | dB (PAM4) | dB (NRZ) | Coefficient Units |
|---|---|---|---|
| 5'b00000 | 0.0 | 0.0 | 0 |
| 5'b00001 | –0.662 | –0.365 | 1 |
| 5'b00010 | –1.03 | –0.590 | 2 |
| 5'b00011 | –1.42 | –0.819 | 3 |
| 5'b00100 | –1.81 | –1.05 | 4 |
| 5'b00101 | –2.22 | –1.29 | 5 |
| 5'b00110 | –2.64 | –1.54 | 6 |
| 5'b00111 | –3.08 | –1.79 | 7 |
| 5'b01000 | –3.50 | –2.04 | 8 |
| 5'b01001 | –3.99 | –2.31 | 9 |
| 5'b01010 | –4.48 | –2.58 | 10 |
| 5'b01011 | –5.02 | –2.87 | 11 |
| 5'b01100 | –5.56 | –3.17 | 12 |
| 5'b01101 | –6.17 | –3.47 | 13 |
| 5'b01110 | –6.80 | –3.77 | 14 |
| 5'b01111 | –7.49 | –4.10 | 15 |
| 5'b10000 | –8.21 | –4.43 | 16 |
| 5'b10001 | –9.03 | –4.78 | 17 |
| 5'b10010 | N/A | –5.15 | 18 |
| 5'b10011 | N/A | –5.53 | 19 |
| 5'b10100 | N/A | –5.93 | 20 |
| 5'b10101 | N/A | –6.35 | 21 |
| 5'b10110 | N/A | –6.77 | 22 |
| 5'b10111 | N/A | –7.24 | 23 |
| 5'b11000 | N/A | –7.73 | 24 |
| 5'b11001 | N/A | –8.25 | 25 |
| 5'b11010 | N/A | –8.51 | 26 |
| 5'b11011 | N/A | –8.74 | 27 |

**Notes:**

1. CH[0/1]_TXEMPPRE de-emphasis values are defined when CH[0/1]_TXEMPPOST = 5'b00000, and CH[0/1]_TXEMPPRE2 = 4'b0000. Emphasis = $20\log10(V_{high}/V_{low}) = |20\log10(V_{low}/V_{high})|$.

*Table 45:* **TX Configurable Driver Ports** *(cont'd)*

<table>
<tr><th>Port</th><th>Dir</th><th>Clock Domain</th><th>Description</th></tr>
<tr><td>CH[0/1]_TXEMPPRE2[3:0]</td><td>Input</td><td>Async</td><td>Transmitter pre-cursor 2 TX pre-emphasis control for channel 0. The default is user specified. All listed values (dB) are typical
<table>
<tr><th>[4:0]</th><th>dB (PAM4)</th><th>dB (NRZ)</th><th>Coefficient Units</th></tr>
<tr><td>4'b0000</td><td>0.0</td><td>0.0</td><td>0</td></tr>
<tr><td>4'b0001</td><td>–0.663</td><td>–0.368</td><td>1</td></tr>
<tr><td>4'b0010</td><td>–1.03</td><td>–0.593</td><td>2</td></tr>
<tr><td>4'b0011</td><td>–1.41</td><td>–0.816</td><td>3</td></tr>
<tr><td>4'b0100</td><td>–1.80</td><td>–1.04</td><td>4</td></tr>
<tr><td>4'b0101</td><td>–2.21</td><td>–1.28</td><td>5</td></tr>
<tr><td>4'b0110</td><td>–2.63</td><td>–1.53</td><td>6</td></tr>
<tr><td>4'b0111</td><td>–3.06</td><td>–1.78</td><td>7</td></tr>
<tr><td>4'b1000</td><td>N/A</td><td>–2.04</td><td>8</td></tr>
<tr><td>4'b1001</td><td>N/A</td><td>–2.30</td><td>9</td></tr>
<tr><td>4'b1010</td><td>N/A</td><td>–2.58</td><td>10</td></tr>
<tr><td>4'b1011</td><td>N/A</td><td>–2.72</td><td>11</td></tr>
<tr><td>4'b1100</td><td>N/A</td><td>–2.88</td><td>12</td></tr>
</table>
Notes:<br>1. CH[0/1]_TXEMPPRE2 de-emphasis values are defined when CH[0/1]_TXEMPPRE = 5'b00000, and CH[0/1]_TXEMPPOST = 4'b0000. Emphasis = 20log10($V_{high}/V_{low}$) = | 20log10($V_{low}/V_{high}$)|.</td></tr>
</table>

*Table 45:* **TX Configurable Driver Ports** *(cont'd)*

<table>
<tr><th>Port</th><th>Dir</th><th>Clock Domain</th><th>Description</th></tr>
<tr><td>CH[0/1]_TXEMPPOST[4:0]</td><td>Input</td><td>Async</td><td>Transmitter post-cursor TX pre-emphasis control for channel 0. The default is user specified. All listed values (dB) are typical.
<table>
<tr><th>[4:0]</th><th>dB (PAM4)</th><th>dB (NRZ)</th><th>Coefficient Units</th></tr>
<tr><td>5'b00000</td><td>0.0</td><td>0.0</td><td>0</td></tr>
<tr><td>5'b00001</td><td>–0.652</td><td>–0.361</td><td>1</td></tr>
<tr><td>5'b00010</td><td>–1.01</td><td>–0.580</td><td>2</td></tr>
<tr><td>5'b00011</td><td>–1.39</td><td>–0.802</td><td>3</td></tr>
<tr><td>5'b00100</td><td>–1.77</td><td>–1.02</td><td>4</td></tr>
<tr><td>5'b00101</td><td>–2.17</td><td>–1.26</td><td>5</td></tr>
<tr><td>5'b00110</td><td>–2.57</td><td>–1.50</td><td>6</td></tr>
<tr><td>5'b00111</td><td>–3.00</td><td>–1.75</td><td>7</td></tr>
<tr><td>5'b01000</td><td>–3.41</td><td>–2.00</td><td>8</td></tr>
<tr><td>5'b01001</td><td>–3.88</td><td>–2.26</td><td>9</td></tr>
<tr><td>5'b01010</td><td>–4.37</td><td>–2.53</td><td>10</td></tr>
<tr><td>5'b01011</td><td>–4.88</td><td>–2.80</td><td>11</td></tr>
<tr><td>5'b01100</td><td>–5.42</td><td>–3.09</td><td>12</td></tr>
<tr><td>5'b01101</td><td>–5.99</td><td>–3.38</td><td>13</td></tr>
<tr><td>5'b01110</td><td>–6.59</td><td>–3.68</td><td>14</td></tr>
<tr><td>5'b01111</td><td>–7.25</td><td>–3.99</td><td>15</td></tr>
<tr><td>5'b10000</td><td>–7.94</td><td>–4.31</td><td>16</td></tr>
<tr><td>5'b10001</td><td>–8.71</td><td>–4.65</td><td>17</td></tr>
<tr><td>5'b10010</td><td>N/A</td><td>–5.00</td><td>18</td></tr>
<tr><td>5'b10011</td><td>N/A</td><td>–5.37</td><td>19</td></tr>
<tr><td>5'b10100</td><td>N/A</td><td>–5.74</td><td>20</td></tr>
<tr><td>5'b10101</td><td>N/A</td><td>–6.14</td><td>21</td></tr>
<tr><td>5'b10110</td><td>N/A</td><td>–6.55</td><td>22</td></tr>
<tr><td>5'b10111</td><td>N/A</td><td>–6.98</td><td>23</td></tr>
<tr><td>5'b11000</td><td>N/A</td><td>–7.46</td><td>24</td></tr>
<tr><td>5'b11001</td><td>N/A</td><td>–7.94</td><td>25</td></tr>
<tr><td>5'b11010</td><td>N/A</td><td>–8.20</td><td>26</td></tr>
<tr><td>5'b11011</td><td>N/A</td><td>–8.46</td><td>27</td></tr>
</table>
Notes:<br>1. CH[0/1]_TXEMPPOST de-emphasis values are defined when TXEMPPRE = 5'b00000, and TXEMPPRE2 = 4'b0000. Emphasis = $20\log10(V_{high}/V_{low}) = |20\log10(V_{low}/V_{high})|$.</td></tr>
<tr><td>CH[0/1]_GTMTXP<br>CH[0/1]_GTMTXN</td><td>Output (pad)</td><td>TX Serial Clock</td><td>Differential complements of one another forming a differential transmit output pair. These ports represent the pads. The locations of these ports must be constrained (see Implementation) and brought to the top of the design.</td></tr>
</table>

*Table 45:* **TX Configurable Driver Ports** *(cont'd)*

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_TXCTLFIRDAT[5:0] | Input | Async | [5:4]: Set to `2'b01` when using the transmitter in 10G-KR mode. For other cases use the value from the Wizard.<br>[3:0]: Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_TXMUXDCDEXHOLD | Input | Async | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_TXMUXDCDORWREN | Input | Async | Reserved. Use the recommended value from the Wizard. |

The following table defines the TX configurable driver attributes.

*Table 46:* **TX Configurable Driver Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_TX_ANA_CFG1 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| TXMODSEL | [7] | Driver output modulation control:<br>`1'b0`: PAM4 modulation.<br>`1'b1`: NRZ modulation. |
| CH[0/1]_TX_DRV_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| TXEMPMAIN_INDEP | [0] | Allows independent control of the main cursor:<br>`1'b0`: The CH[0/1]_TXEMPMAIN coefficient is automatically determined.<br>`1'b1`: CH[0/1]_TXEMPMAIN coefficient can be independently set by the CH[0/1]_TXEMPMAIN pins within the range specified in the pin description. |

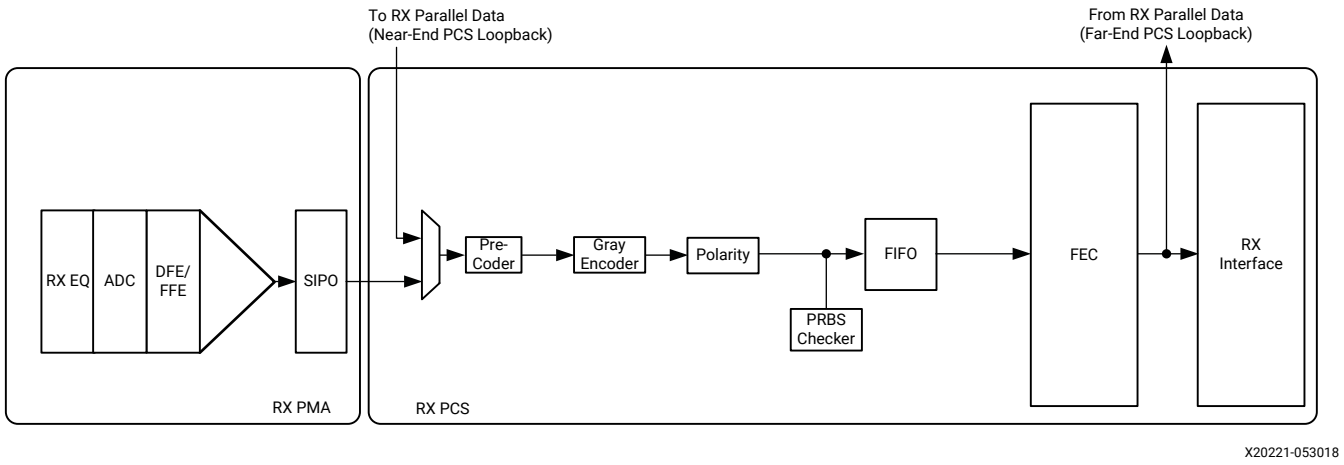
# Use Modes

The GTM TX has the ability to transmit serial data using two different modulation schemes: NRZ and PAM4. NRZ signals contains one bit of information per symbol, while PAM4 signals contain two bits of information per symbol. Using PAM4 modulation doubles the transmitted data bandwidth while maintaining the same unit interval (UI). To program the GTM TX to a desired signal modulation mode, the user must configure the attribute TXMODSEL for CH0 or CH1.

# Receiver

This section shows how to configure and use each of the functional blocks inside the receiver (RX). Each GTM transceiver includes an independent receiver made up of a PCS and PMA. The following figure shows the blocks of the GTM transceiver RX. High-speed serial data flows from traces on the board into the PMA of the GTM transceiver RX, into the PCS, and finally into the interconnect logic.

*Figure 37:* **GTM Transciever RX Block Diagram**




The key elements within the GTM transceiver RX are:

1. RX Analog Front End
2. RX Equalizer
3. RX CDR
4. RX Fabric Clock Output Control
5. RX Margin Analysis
6. RX Pre-Coder
7. RX Gray Encoder
8. RX Polarity Control
9. RX Pattern Checker
10. RX Buffer

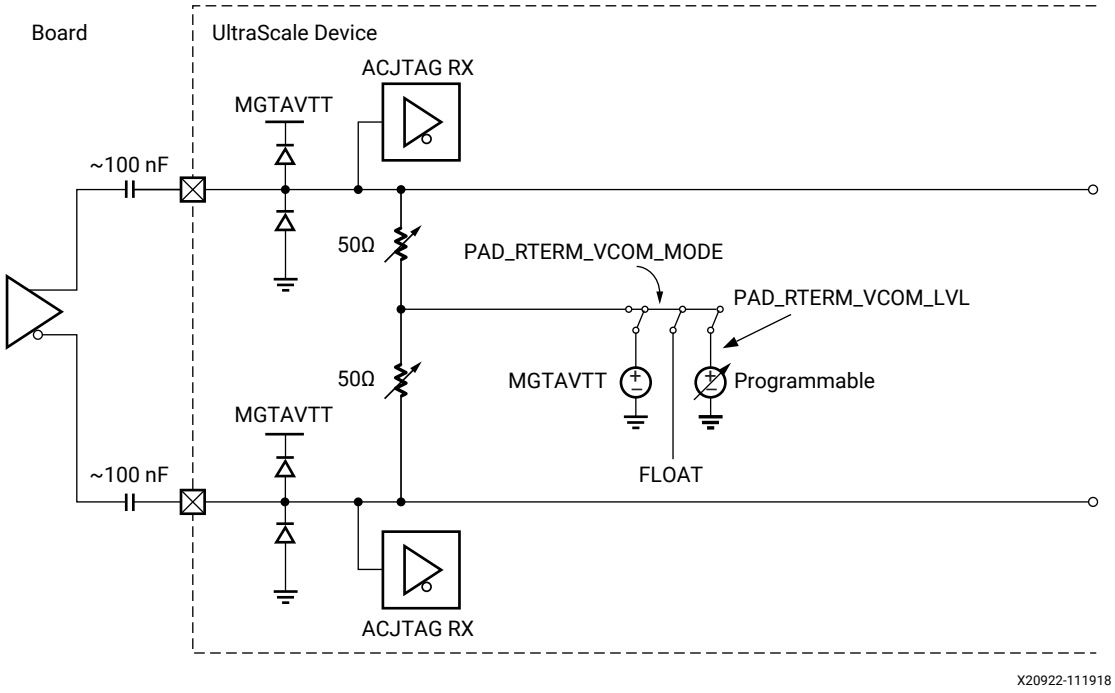
11. RX FEC

# RX Analog Front End

The RX analog front end (AFE) is an ADC-based input differential buffer. It has the following features:

- Configurable RX termination voltage
- Calibrated termination resistors

*Figure 38:* **RX Analog Front End**




## Ports and Attributes

The following table defines the RX AFE ports.

*Table 47:* **RX AFE ports**

| Ports | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_GTMRXP, CH[0/1]_GTMRXN | In (Pad) | RX Serial Clock | Differential complements of one another forming a differential receiver input pair. These ports represent pads. The location of these ports must be constrained (see Implementation) and brought to the top level of the design. |

*Table 47:* **RX AFE ports** *(cont'd)*

| Ports | Dir | Clock Domain | Description |
|---|---|---|---|
| BGRCALOVRD[4:0] | In | Async | Reserved. This port must be set to `5'b00000`. Do not modify this value. |
| BGRCALOVRDENB | In | Async | Reserved. This port must be set to `1'b1`. Do not modify this value. |
| RCALENB | In | Async | Reserved. This port must be set to `1'b1`. Do not modify this value. |

The following table defines the RX AFE attributes.

*Table 48:* **RX AFE Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RX_PAD_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| PAD_RTERM_VCOM_MODE | [12:11] | Controls the mode for the RX termination voltage:<br>`2'b00`: AVTT<br>`2'b01`: Reserved<br>`2'b10`: Floating<br>`2'b11`: Programmable |
| PAD_RTERM_VCOM_LVL | [10:7] | Controls the common mode in programmable mode:<br>`4'b0000`: 100 mV<br>`4'b0001`: 200 mV<br>`4'b0010`: 250 mV<br>`4'b0011`: 300 mV<br>`4'b0100`: 350 mV<br>`4'b0101`: 400 mV<br>`4'b0110`: 500 mV<br>`4'b0111`: 550 mV<br>`4'b1000`: 600 mV<br>`4'b1001`: 700 mV<br>`4'b1010`: 800 mV<br>`4'b1011`: 850 mV<br>`4'b1100`: 900 mV<br>`4'b1101`: 950 mV<br>`4'b1110`: 1000 mV<br>`4'b1111`: 1100 mV |

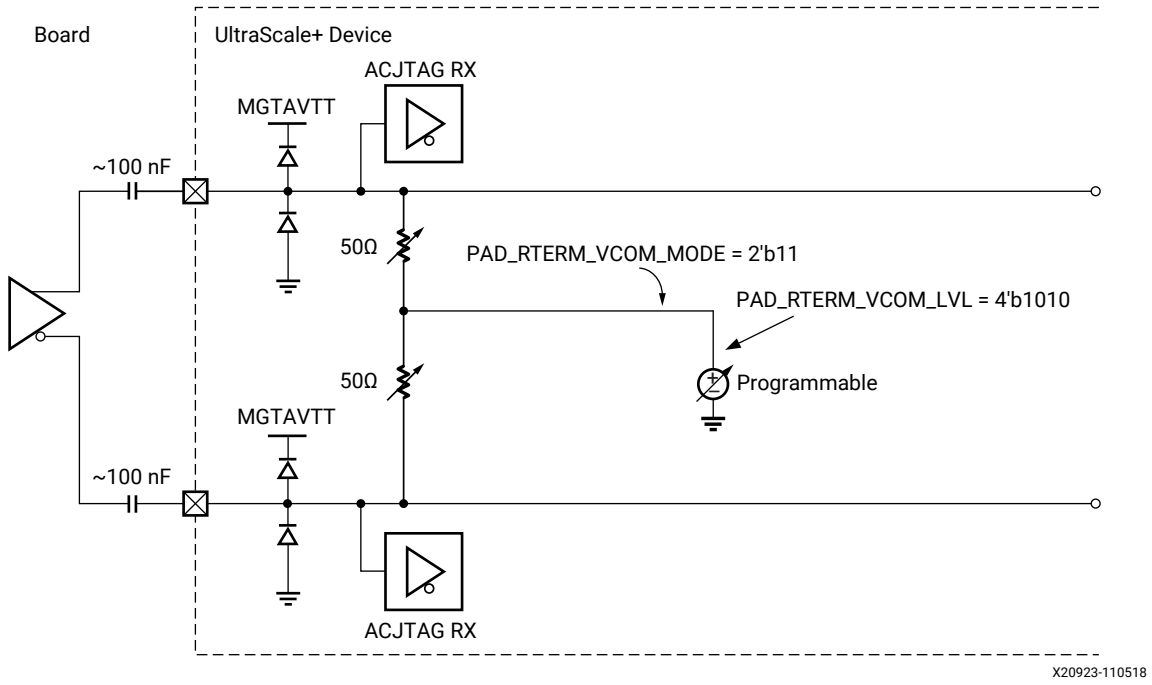
# GTM Use Modes - RX Termination

This section describes the GTM use modes with RX termination.

*Table 49:* **GTM Use Mode 1 - RX Termination**

| Use Mode | External AC Coupling | Term Voltage (mV) | Suggested Protocols and Usage Notes |
|---|---|---|---|
| 1 | On | 800 | Attribute settings:<br>PAD_RTERM_VCOM_MODE = `2'b11`<br>PAD_RTERM_VCOM_LVL = `4'b1010` |

*Figure 39:* **Use Mode 1**

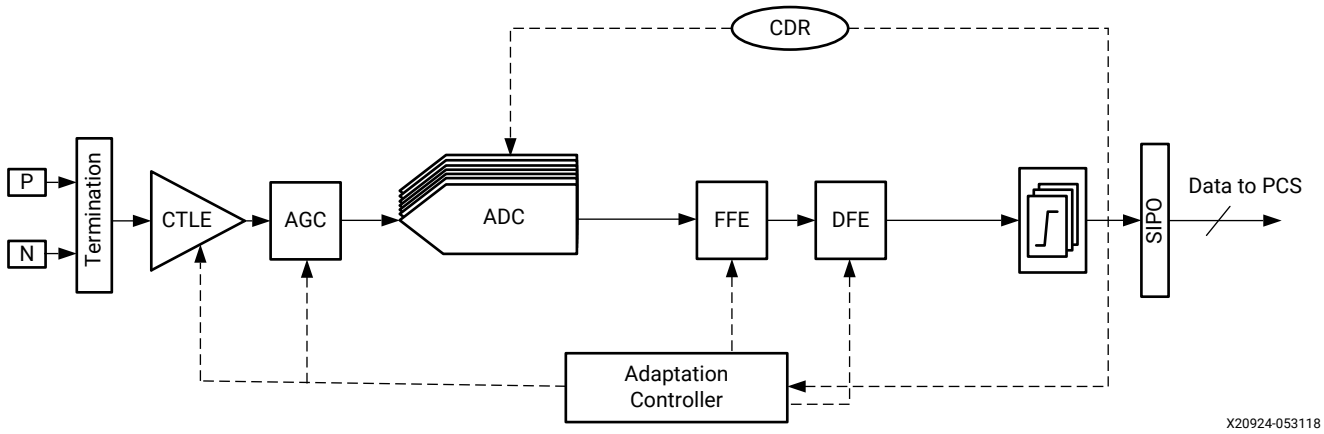



# RX Equalizer

A serial link bit error rate (BER) performance is a function of the transmitter, transmission media, and receiver. The transmission media of the channel is bandwidth-limited, and the signal traveling through is subjected to attenuation and distortion.

The GTM receiver is an ADC-based buffer that breaks the equalizer into two domains: analog and digital. The incoming signal first passes through the analog stage consisting of a CTLE and AGC stage. The signal is then digitized by the ADC, and passes through the Feed-Forward Equalizer (FFE) and Decision-Feedback-Equalizer (DFE), as shown in the following figure.

*Figure 40:* **GTM RX Equalization**




The combination of CTLE, FFE, and DFE can compensate for both the pre-cursor and post-cursor of the transmitted bit. All equalization loops are auto-adaptive to handle a wide range of channel profiles and to compensate for any PVT variations.

# Ports and Attributes

The following table defines the RX equalizer ports.

*Table 50:* **RX Equalizer Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_RXADAPTRESET | In | Async | This port is driven High and then deasserted to start a single-mode reset on RX adaptation. The reset is not dependent on RXRESETMODE or RXPMARESETMASK setting. |
| CH[0/1]_RXADCCALRESET | In | Async | Reserved. Tie to `1'b0`. |
| CH[0/1]_RXADCCLKGENRESET | In | Async | This port is driven High and then deasserted to start a single-mode reset on the RX ADC CLKGEN. The reset is not dependent on RXRESETMODE or RXPMARESETMASK setting. |
| CH[0/1]_RXDFERESET | In | Async | This port is driven High and then deasserted to start a single-mode reset on the DFE. The reset is not dependent on RXRESETMODE or RXPMARESETMASK setting. |
| CH[0/1]_RXDSPRESET | In | Async | This port is driven High and then deasserted to start a single-mode reset on the DSP. The reset is not dependent on RXRESETMODE or RXPMARESETMASK setting. |
| CH[0/1]_RXEQTRAINING | In | Async | Reserved. Tie to `1'b0`. |

The following table defines the RX equalizer attributes.

*Table 51:* **RX Equalizer Attributes**

| Attribute | Type | Description | |
|---|---|---|---|
| CH[0/1]_RX_APT_CTRL_CFG2 | 16-bit | Reserved. | |
| **Bit Name** | **Address** | **Description** | |
| RXMODSEL | [3] | Receiver input modulation control.<br>`0'b0`: PAM4 modulation.<br>`0'b1`: NRZ modulation. | |
| CH[0/1]_RX_APT_CFG27A[15:0] | 16-bit | Adaptation loop freeze controls. Use the recommended value from the Wizard. Freezing DFE Tap 1 (H1) is not supported. | |
| | | **Bit** | **Description** |
| | | [15:13] | Reserved. |
| | | [12] | Enable to freeze current automatic gain control (AGC) adapt value. |
| | | [11] | Enable to freeze current CTLE High frequency loop (KH) adapt value. |
| | | [10] | Enable to freeze current CTLE Low frequency loop (KL) adapt value. |
| | | [9] | Enable to freeze current Offset Cancellation (OS) adapt value. |
| | | [8:5] | Reserved. |
| | | [4] | Enable to freeze current FFE Pre-Cursor Tap HM04 adapt value. |
| | | [3] | Enable to freeze current FFE Pre-Cursor Tap HM03 adapt value. |
| | | [2] | Enable to freeze current FFE Pre-Cursor Tap HM02 adapt value. |
| | | [1] | Enable to freeze current FFE Pre-Cursor Tap HM01 adapt value. |
| | | [0] | Adaptation freeze control. This bit must be enabled to freeze the loops selected in CH[0/1]_RX_APT_CFG27A, and CH[0/1]_RX_APT_CFG27B. If this bit is set to Low, all loops will be auto-adapting. |

*Table 51:* **RX Equalizer Attributes** *(cont'd)*

<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>CH[0/1]_RX_APT_CFG27B[15:0]</td><td>16-bit</td><td>Adaptation loop freeze controls. Use the recommended value from the Wizard. Freezing DFE Tap 1 (H1) is not supported.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>[15:9]</td><td>Reserved.</td></tr>
<tr><td>[8]</td><td>Enable to freeze current FFE Post-Cursor Tap HP10 adapt value.</td></tr>
<tr><td>[7]</td><td>Enable to freeze current FFE Post-Cursor Tap HP09 adapt value.</td></tr>
<tr><td>[6]</td><td>Enable to freeze current FFE Post-Cursor Tap HP08 adapt value.</td></tr>
<tr><td>[5]</td><td>Enable to freeze current FFE Post-Cursor Tap HP07 adapt value.</td></tr>
<tr><td>[4]</td><td>Enable to freeze current FFE Post-Cursor Tap HP06 adapt value.</td></tr>
<tr><td>[3]</td><td>Enable to freeze current FFE Post-Cursor Tap HP05 adapt value.</td></tr>
<tr><td>[2]</td><td>Enable to freeze current FFE Post-Cursor Tap HP04 adapt value.</td></tr>
<tr><td>[1]</td><td>Enable to freeze current FFE Post-Cursor Tap HP03 adapt value.</td></tr>
<tr><td>[0]</td><td>Enable to freeze current FFE Post-Cursor Tap HP02 adapt value.</td></tr>
</table>
***Note***: Attribute CH[0/1]_RX_APT_CFG27A[0] must be enabled to freeze the enabled loops in CH[0/1]_RX_APT_CFG27B[15:0].</td></tr>
</table>

*Table 51:* **RX Equalizer Attributes** *(cont'd)*

| Attribute | Type | Description | |
|---|---|---|---|
| CH[0/1]_RX_APT_CFG28A[15:0] | 16-bit | Adaptation loop override controls. Use the recommended value from the Wizard. Overriding DFE Tap 1 (H1) is not supported. | |
| | | **Bit** | **Description** |
| | | [15:13] | Reserved. |
| | | [12] | Enable to override automatic gain control (AGC) value according to attribute CH[0/1]_RX_APT_CFG18A[11:6]. |
| | | [11] | Enable to override CTLE High frequency loop (KH) value according to attribute CH[0/1]_RX_APT_CFG18A[5:0]. |
| | | [10] | Enable to override CTLE Low frequency loop (KL) value according to attribute CH[0/1]_RX_APT_CFG17B[12:7]. |
| | | [9] | Enable to override Offset Cancellation (OS) value according to attribute CH[0/1]_RX_APT_CFG17B[6:0]. |
| | | [8:5] | Reserved. |
| | | [4] | Enable to override FFE Pre-Cursor Tap HM04 value according to attribute CH[0/1]_RX_APT_CFG14B[5:0]. |
| | | [3] | Enable to override FFE Pre-Cursor Tap HM03 value according to attribute CH[0/1]_RX_APT_CFG14B[11:6]. |
| | | [2] | Enable to override FFE Pre-Cursor Tap HM02 value according to attribute CH[0/1]_RX_APT_CFG15A[13:7]. |
| | | [1] | Enable to override FFE Pre-Cursor Tap HM01 value according to attribute CH[0/1]_RX_APT_CFG14A[15:8]. |
| | | [0] | Adaptation override control. This bit must be enabled to override the loops selected in CH[0/1]_RX_APT_CFG28A and CH[0/1]_RX_APT_CFG28B. If this bit is set to Low, all loops will be auto-adapting. |

*Table 51:* **RX Equalizer Attributes** *(cont'd)*

<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>CH[0/1]_RX_APT_CFG28B[15:0]</td><td>16-bit</td><td>Adaptation loop override controls. Use the recommended value from the Wizard. Overriding DFE Tap 1 (H1) is not supported.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>[15:9]</td><td>Reserved.</td></tr>
<tr><td>[8]</td><td>Enable to override FFE Post-Cursor Tap HP10 value according to attribute CH[0/1]_RX_APT_CFG12B[9:5].</td></tr>
<tr><td>[7]</td><td>Enable to override FFE Post-Cursor Tap HP09 value according to attribute CH[0/1]_RX_APT_CFG12B[14:10].</td></tr>
<tr><td>[6]</td><td>Enable to override FFE Post-Cursor Tap HP08 value according to attribute CH[0/1]_RX_APT_CFG13A[4:0].</td></tr>
<tr><td>[5]</td><td>Enable to override FFE Post-Cursor Tap HP07 value according to attribute CH[0/1]_RX_APT_CFG13A[9:5].</td></tr>
<tr><td>[4]</td><td>Enable to override FFE Post-Cursor Tap HP06 value according to attribute CH[0/1]_RX_APT_CFG13A[15:10].</td></tr>
<tr><td>[3]</td><td>Enable to override FFE Post-Cursor Tap HP05 value according to attribute CH[0/1]_RX_APT_CFG13B[5:0].</td></tr>
<tr><td>[2]</td><td>Enable to override FFE Post-Cursor Tap HP04 value according to attribute CH[0/1]_RX_APT_CFG13B[12:6].</td></tr>
<tr><td>[1]</td><td>Enable to override FFE Post-Cursor Tap HP03 value according to attribute CH[0/1]_RX_APT_CFG15A[6:0].</td></tr>
<tr><td>[0]</td><td>Enable to override FFE Post-Cursor Tap HP02 value according to attribute CH[0/1]_RX_APT_CFG14A[7:0].</td></tr>
</table>
***Note***: Attribute CH[0/1]_RX_APT_CFG28A[0] must be enabled to override the enabled loops in CH[0/1]_RX_APT_CFG28B[15:0].</td></tr>
</table>

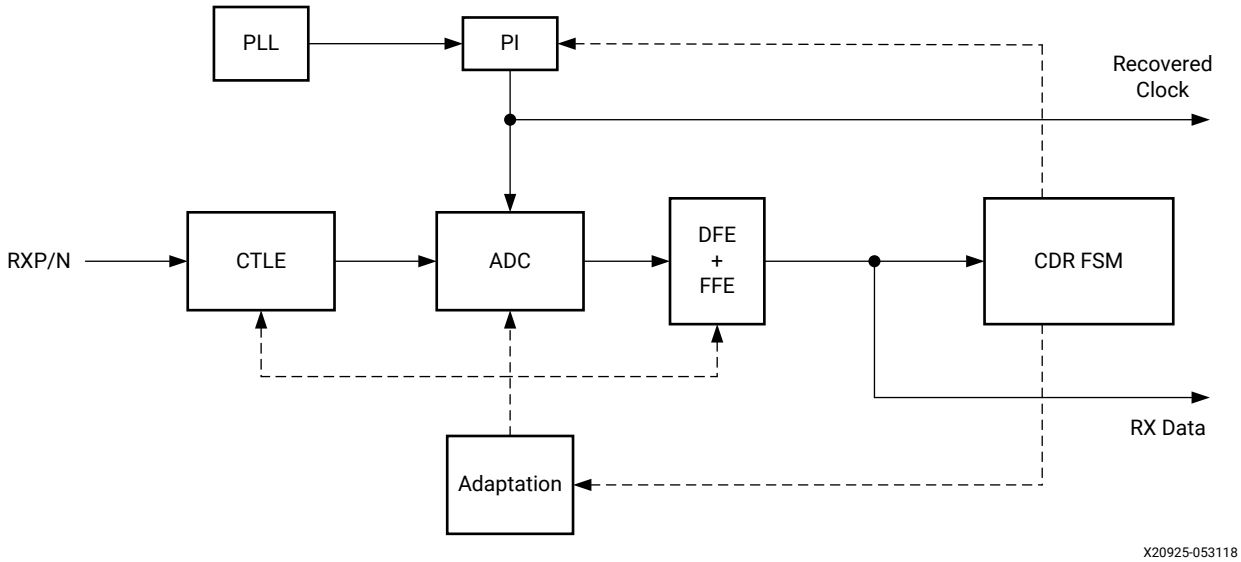
# Use Modes

The GTM RX has the ability to receive serial data using two different modulation schemes: NRZ and PAM4. NRZ signals contain one bit of information per symbol, while PAM4 signals contain two bits of information per symbol. Using PAM4 modulation doubles the transmitted data bandwidth while maintaining the same unit interval (UI). To program the GTM RX to a desired signal modulation mode, the user must configure the attribute RXMODSEL for CH0 or CH1.

# RX CDR

The RX clock data recovery (CDR) circuit in each UltraScale+ FPGA GTM transceiver channel extracts the recovered clock and data from an incoming data stream. The following figure illustrates the architecture of the CDR block. Clock paths are shown with dotted lines for clarity.

*Figure 41:* **CDR Block Diagram**




The GTM transceiver employs the baud-rate phase detection CDR architecture. Incoming data first goes through receiver equalization and ADC where the data is sampled. The sampled data then moves through FFE and DFE before feeding to the CDR state machine and the downstream transceiver blocks.

The LCPLL provides a base clock to the phase interpolator. The phase interpolator in turn produces fine, evenly spaced sampling phases to allow the CDR state machine to have fine phase control. The CDR state machine can track incoming data streams that can have a frequency offset from the local PLL reference clock.

## Ports and Attributes

The following table defines the CDR ports.

*Table 52:* **CDR Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_RXCDROVRDEN | In | Async | Reserved. Use the recommended value from the Wizard. |

*Table 52:* **CDR Ports** *(cont'd)*

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_RXCDRFRRESET | In | Async | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_RXCDRHOLD | In | Async | Hold the CDR control loop frozen. |
| CH[0/1]_RXCDRINCPCTRL | In | Async | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_RXCDRPHRESET | In | Async | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_RXCDRFREQOS | In | Async | Reserved. Use the recommended value from the Wizard. |

The following table defines the CDR attributes.

*Table 53:* **CDR Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RX_CDR_CFG0A | 16-bit | CDR configuration. Use the recommended value from the Wizard. |
| CH[0/1]_RX_CDR_CFG0B | 16-bit | CDR configuration. Use the recommended value from the Wizard. |
| CH[0/1]_RX_CDR_CFG1A | 16-bit | CDR configuration. Use the recommended value from the Wizard. |
| CH[0/1]_RX_CDR_CFG1B | 16-bit | CDR configuration. Use the recommended value from the Wizard. |
| CH[0/1]_RX_CDR_CFG2A | 16-bit | CDR configuration. Use the recommended value from the Wizard. |
| CH[0/1]_RX_CDR_CFG2B | 16-bit | CDR configuration. Use the recommended value from the Wizard. |
| CH[0/1]_RX_CDR_CFG3A | 16-bit | CDR configuration. Use the recommended value from the Wizard. |
| CH[0/1]_RX_CDR_CFG3B | 16-bit | CDR configuration. Use the recommended value from the Wizard. |
| CH[0/1]_RX_CDR_CFG4A | 16-bit | CDR configuration. Use the recommended value from the Wizard. |
| CH[0/1]_RX_CDR_CFG4B | 16-bit | CDR configuration. Use the recommended value from the Wizard. |

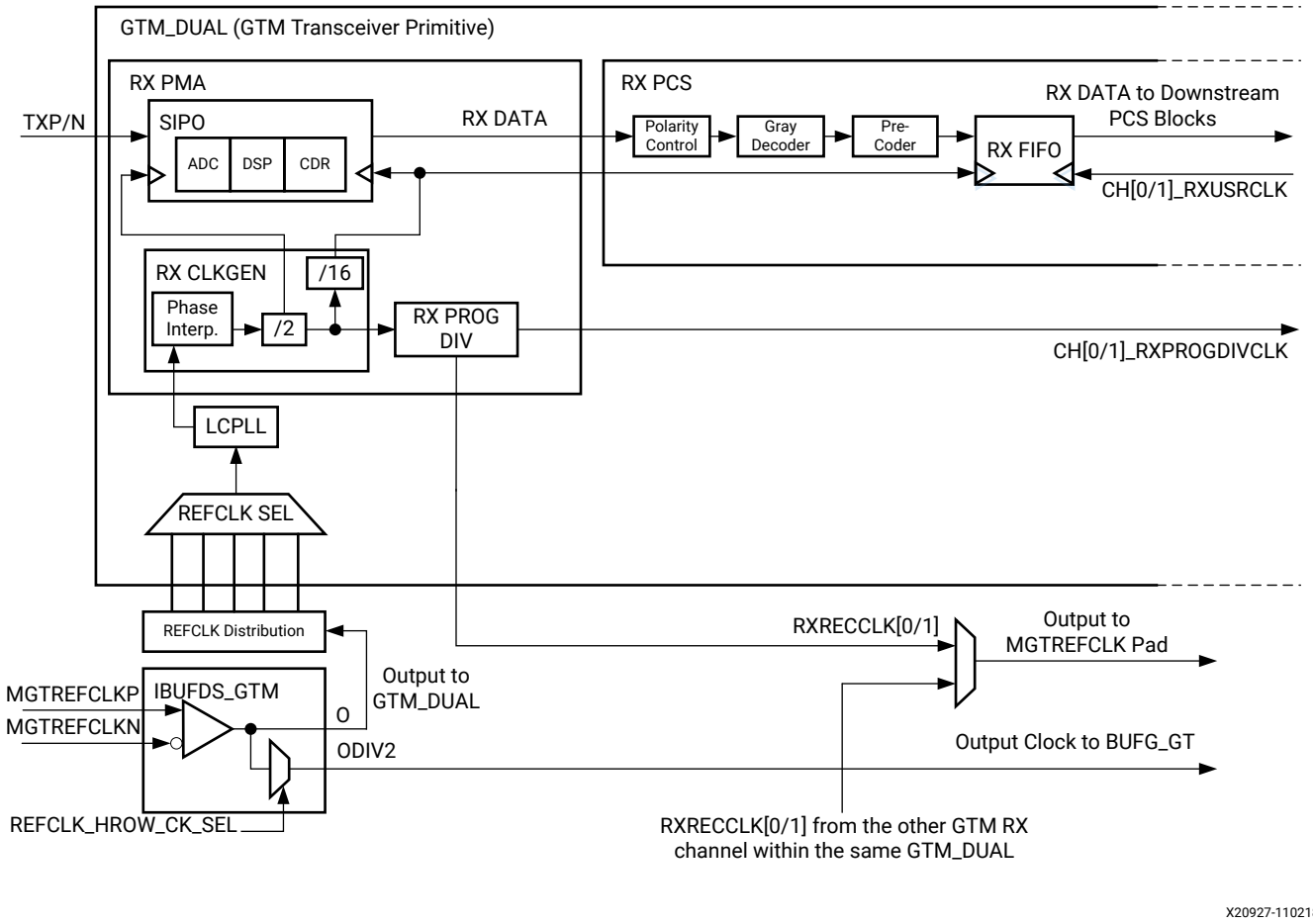
# RX CDR Lock to Reference

To get the CDR to lock to reference, set CH[0/1]_RXCDRHOLD = `1'b1` and CH[0/1]_RXCDROVRDEN = `1'b0`.

# RX Fabric Clock Output Control

The RX Clock Divider Control block has two main components: serial clock divider control, and parallel clock divider and selector control. The clock divider and selector details are illustrated in the following figure.

*Figure 42:* **RX Serial and Parallel Clock Divider**




Notes relevant to the figure:

1. CH[0/1]_RXPROGDIVCLK is used as the source of the interconnect logic clock via BUFG_GT.
2. There is only one LCPLL in the GTM_DUAL primitive which is shared between the TX/RX.
3. RXRECCLK[0/1] is the same as CH[0/1]_RXPROGDIVCLK. It can be routed to the MGTREFCLK output pad to be used elsewhere.

# RX Programmable Divider

The RX programmable divider shown in Figure 42 uses the LCPLL output clock to generate a parallel output clock. By using the transceiver LCPLL, RX programmable divider, and BUFG_GT, CH[0/1]_RXPROGDIVCLK should be used as a clock source for the interconnect logic.

The following tables show the programmable divider ports and attributes, respectively.

*Table 54:* **RX Programmable Divider Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_RXPROGDIVRESET | In | Async | This active-High port resets the dividers as well as the RXPRGDIVRESETDONE indicator. A reset must be performed whenever the input clock source is interrupted. |
| CH[0/1]_RXPRGDIVRESETDONE | Out | Async | When the input clock is stable and reset is performed, this active-High signal indicates the reset is completed and the output clock is stable. |
| CH[0/1]_RXPROGDIVCLK | Out | Clock | RXPROGDIVCLK is the parallel clock output from The RX programmable divider. This clock is the recommended output to the interconnect logic through BUFG_GT. |

*Table 55:* **RX Programmable Divider Attribute**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RX_ANA_CFG1 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| RX_PROGDIV_SELFR | [13] | This attribute is used during the RX programmable divider ratio selection. Set to `1'b1` to obtain the full rate of the divided clock. Set to `1'b0` to obtain the half rate of the divided clock. |
| RX_PROGDIV_SEL_DIV66 | [12] | This attribute is used during the RX programmable divider ratio selection.<br>• The attribute must be set to `1'b1` when the desired divider value is either 16.5, 33, 66, or 132.<br>• For all other divider values, this should be set to `1'b0`. |
| RX_PROGDIV_SEL_DIV5 | [11] | This attribute is used during the RX programmable divider ratio selection.<br>• The attribute must be set to `1'b0` when the desired divider value is either 16.5, 33, 66, or 132.<br>• For all other divider values, this should be set to `1'b1`. |

*Table 55:* **RX Programmable Divider Attribute** *(cont'd)*

| Attribute | Type | Description |
|---|---|---|
| RX_PROGDIV_FBDIV | [7:2] | This attribute is the main RX programmable divider selector.<br>When the following settings are set:<br>RX_PROGDIV_SEL_DIV66 = `1'b0`<br>RX_PROGDIV_SEL_DIV5 = `1'b1`<br>RX_PROGDIV_SELFR = `1'b1` (or `1'b0`)<br>Valid RX programmable divider ratios are:<br>`6'b011000`: 4 (8)<br>`6'b111000`: 5 (10)<br>`6'b000000`: 8 (16)<br>`6'b100000`: 10 (20)<br>`6'b000001`: 12 (24)<br>`6'b100001`: 15 (30)<br>`6'b000010`: 16 (32)<br>`6'b100010`: 20 (40)<br>`6'b000101`: 24 (48)<br>`6'b100011`: 25 (50)<br>`6'b100101`: 30 (60)<br>`6'b000110`: 32 (64)<br>`6'b100110`: 40 (80)<br>`6'b001101`: 48 (96)<br>`6'b100111`: 50 (100)<br>`6'b101101`: 60 (120)<br>`6'b001110`: 64 (128)<br>`6'b001111`: 80 (160)<br>`6'b101111`: 100 (200)<br>When the following settings are set:<br>RX_PROGDIV_SEL_DIV66 = `1'b1`<br>RX_PROGDIV_SEL_DIV5 = `1'b0`<br>RX_PROGDIV_SELFR = `1'b1` (or `1'b0`)<br>Valid RX programmable divider ratios are:<br>`6'b011000`: 16.5 (33)<br>`6'b000000`: 33 (66)<br>`6'b000010`: 66 (132) |

# Ports and Attributes

The following table defines the ports required for TX fabric clock output control.

*Table 56:* **RX Fabric Clock Output Control Ports**

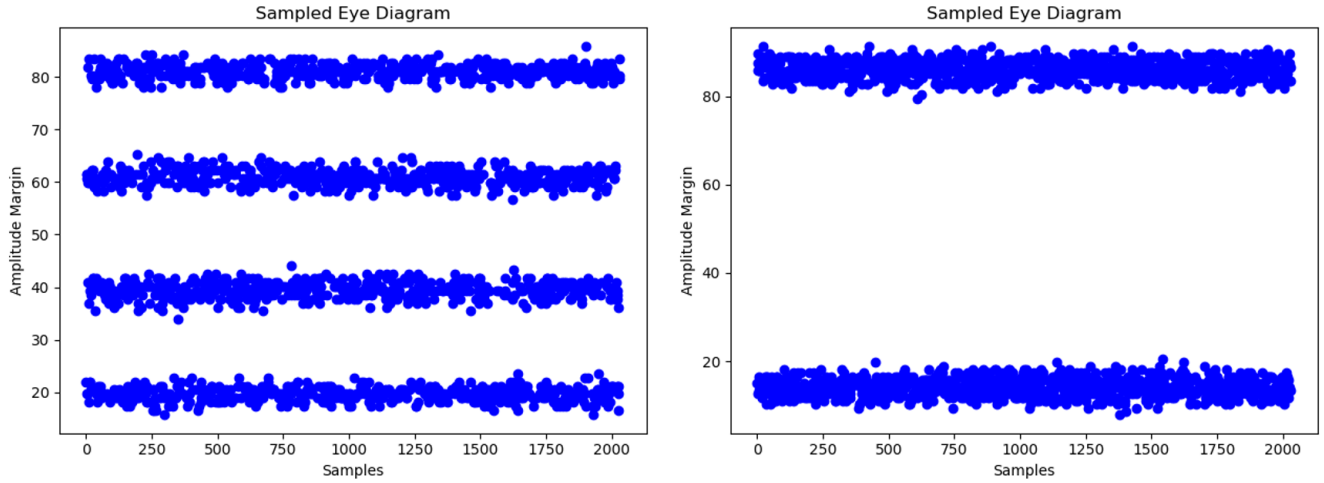
| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_RXOUTCLKSEL[2:0] | In | Async | This port must be set to `3'b000`. |
| CH[0/1]_RXOUTCLK | Out | Clock | Reserved. |
| CH[0/1]_RXPROGDIVCLK | Out | Clock | RXPROGDIVCLK is the parallel output clock from the RX programmable divider. This clock is the recommended output to the interconnect logic through BUFG_GT. |
| RXRECCLK[0/1] | Out | Clock | RXRECCLK is the same as RXPROGDIVCLK, and it can be routed to the MGTREFCLK output pad. |

# RX Margin Analysis

As line rates and channel attenuation increase, the receiver equalizers are more often enabled to overcome channel attenuation. This poses a challenge to system bring-up because the quality of the link cannot be determined by measuring the far-end eye opening at the receiver pins. At high line rates, the received eye measurement on the printed circuit board can appear to be completely closed even though the internal eye after the receiver equalizer is open.

Because the GTM receiver is ADC-based, the conventional eye scan used in the previous family of transceivers (such as GTH and GTY transceivers) is not possible. The GTM RX provides a non-destructive sampled eye diagram that can be used to measure and visualize the receiver signal margin after the equalizer, as shown in the following figure.

*Figure 43:* **Sampled Eye Diagram for (a) PAM4 and (b) NRZ Modulation**



The sampled eye diagram is constructed by plotting one sample per symbol located at the CDR sampling point (after the equalizer). By plotting multiple samples, the image looks like the figure above. Sampled eye supports both NRZ and PAM4 modulation, with the only difference being that the NRZ eye consists of two amplitude margin levels, and PAM4 with four distinct amplitude margin levels. The GT Wizard IP allows access to the sampled eye data. See *Virtex UltraScale+ FPGAs GTM Transceivers Wizard LogiCORE IP Product Guide* (PG315).

# RX Pre-Coder

UltraScale+ GTM receiver supports pre-coding. Pre-coding can be used to reduce receiver DFE error propagation by reducing 1-tap burst error runs into two errors for every error event.

## Ports and Attributes

The following table defines the attributes required for RX pre-coder control.

*Table 57:* **Pre-Coder Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RX_PCS_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| RX_PRECODE_ENDIAN | [11] | In PAM4 mode, this attribute controls pre-coder received endianness. In NRZ mode, the default Wizard value must be used.<br>`1'b0`: Big endian.<br>`1'b1`: Little endian. |
| RX_PRECODE_BYP_EN | [10] | In PAM4 mode, this attribute enables pre-coding. In NRZ mode, the default Wizard value must be used.<br>`1'b0`: Enables pre-code.<br>`1'b1`: Disables pre-code. |

**IMPORTANT!** *In PAM4 mode, if the pre-coder is enabled for the receiver, the transmitter pre-coder should also be enabled for proper data recovery.*

# RX Gray Encoder

The UltraScale+ FPGA GTM receiver supports two types of binary enconding options: linear and Gray coding. By using Gray coding, only one bit error per symbol is made for incorrect decisions, thus reducing the bit-error rate by more than 33%. Table 58 illustrates the differences between linear and Gray Coding.

## Ports and Attributes

The following table defines the attributes required for RX Gray Encoder control.

*Table 58:* **Gray Encoder Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RX_PCS_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| RX_GRAY_ENDIAN | [13] | In PAM4 mode, this attribute controls the received endianness. In NRZ mode, the default Wizard value must be used.<br>`1'b0`: Big endian.<br>`1'b1`: Little endian. |
| RX_GRAY_BYP_EN | [12] | In PAM4 mode, this attribute enables Gray encoding. In NRZ mode, the default Wizard value must be used.<br>`1'b0`: Enables Gray encoding.<br>`1'b1`: Disables Gray encoding. |

**IMPORTANT!** *In PAM4 mode, if the Gray Encoder is enabled for the receiver, the transmitter Gray Encoder should also be enabled for proper data recovery.*

# RX Polarity Control

If the RXP and RXN differential traces are accidentally swapped on the PCB, the differential data received by the GTM RX is reversed. The GTM RX allows inversion to be done on parallel bytes in the PCS after the SIPO to offset reversed polarity on the differential pair. The polarity control function uses the CH0_RXPOLARITY and CH1_RXPOLARITY input, which is driven High from the interconnect logic interface to invert polarity.

## Ports and Attributes

The following table defines the ports required for RX polarity control.

*Table 59:* **RX Polarity Control Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_RXPOLARITY[1] | In | CH[0/1]_RXUSRCLK2 | The CH[0/1]_RXPOLARITY port can invert the polarity of incoming data:<br>0: Not inverted. RXP is positive, and RXN is negative.<br>1: Inverted. RXP is negative, and RXN is positive. |

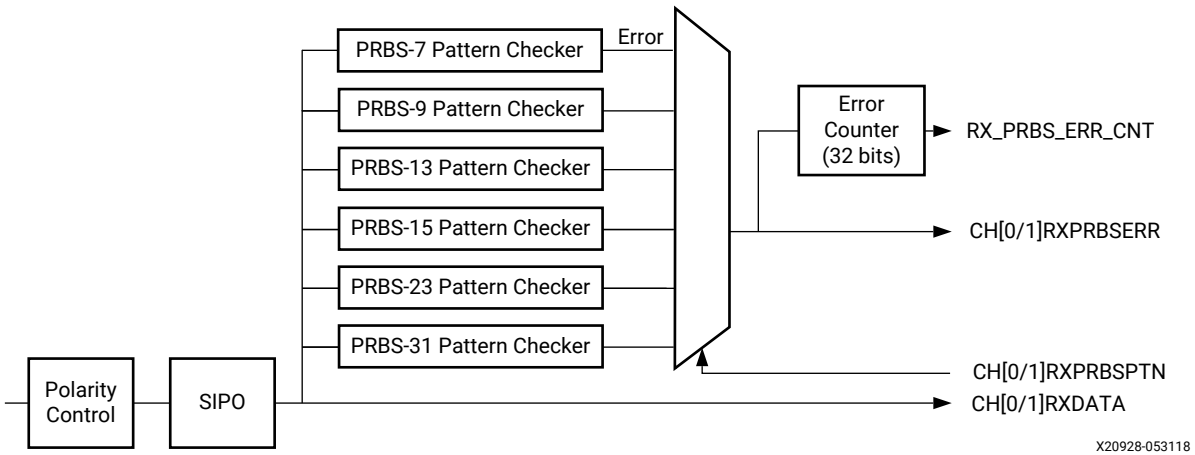
**Notes:**

1. CH[0/1]_RXPOLARITY can be tied High if the polarity of RXP and RXN needs to be reversed.

# RX Pattern Checker

The GTM receiver includes a built-in PRBS checker (see the following figure). This checker can be set to check for one of the six industry-standard PRBS patterns. The checker is self-synchronizing and works on the incoming data before comma alignment or decoding. This function can be used to test the signal integrity of the channel.

*Figure 44:* **RX Pattern Checker Block**




## Ports and Attributes

The following table defines the pattern checker ports.

*Table 60:* **Pattern Checker Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_RXPRBSCSCNTRST | In | CH[0/1]_RXUSRCLK2 | Reset the PRBS error counter. |
| CH[0/1]_RXPRBSPTN[3:0] | In | CH[0/1]_RXUSRCLK2 | Receiver PRBS checker test pattern control. Only these settings are valid:<br>`4'b0000`: Standard operation mode (PRBS check is off)<br>`4'b0001`: PRBS-7<br>`4'b0010`: PRBS-9<br>`4'b0011`: PRBS-15<br>`4'b0100`: PRBS-23<br>`4'b0101`: PRBS-31<br>`4'b0110`: PRBS-13<br>After changing patterns, perform a reset of the RX or a reset of the PRBS error counter (CH[0/1]_RXPRBSCSCNTRST) such that the RX pattern checker can attempt to reestablish the link acquired. No checking is done for non-PRBS patterns. |
| CH[0/1]_RXPRBSERR | Out | CH[0/1]_RXUSRCLK2 | This non-sticky status output indicates that PRBS errors have occurred. Only use CH[0/1]_RX_PRBSERR to read the precise bit error counts. |
| CH[0/1]_RXPRBSLOCKED | Out | CH[0/1]_RXUSRCLK2 | Output to indicate that the RX PRBS checker has been error free for CH[0/1]_RX_PCS_CFG1[12:5] cycles after reset. After being asserted High, CH[0/1]_RXPRBSLOCKED does not deassert until reset of the RX pattern checker via a reset of the RX (GTRXRESET, RXPMARESET, or RXPCSRESET in sequential mode) or a reset of the PRBS error counter (CH[0/1]_RXPRBSCSCNTRST). |
| CH[0/1]_RXPRBSCNTSTOP | In | CH[0/1]_RXUSRCLK2 | Input control to stop the error counter. The PRBS checker needs to restart the error counter after a PRBS stop. |

The following table defines the pattern checker attributes.

*Table 61:* **Pattern Checker Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_TX_PCS_CFG1 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| RXPRBSERR_LOOPBACK | [7] | When this attribute is set to 1, the CH[0/1]_RXPRBSERR bit is internally looped back to CH[0/1]_TXPRBSINERR of the same GTM transceiver. This allows synchronous and asynchronous jitter tolerance testing without worrying about data clock domain crossing. When this attribute is set to `1b'0`, CH[0/1]_TXPRBSINERR is forced onto the TX PRBS. |

*Table 61:* **Pattern Checker Attributes** *(cont'd)*

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RX_PCS_CFG1 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| RXPRBS_LINKACQ_CNT | [12:5] | RX pattern checker link acquire count. Used in conjunction with output port CH[0/1]_RXPRBSLOCKED. After the RX PRBS checker has seen CH[0/1]_RX_PCS_CFG1[12:5] XCLK cycles of error-free PRBS data, CH[0/1]_RXPRBSLOCKED is asserted High. The valid range is 15–255. |

The following table defines the pattern checker DRP read-only registers.

*Table 62:* **Pattern Checker DRP Read-Only Registers**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RX_PRBSERR | 32-bit Binary | PRBS error counter. This counter can be reset by asserting CH[0/1]_RXPRBSCSCNTRST. When a single bit error occurs in incoming data, this counter increments by 1. Single bit errors are thus counted when multiple bit errors occur in incoming data. The counter increments by the actual number of bit errors. Counting begins after RXPRBSLOCKED is asserted High. The counter saturates at `32'hFFFFFFFF`. This error counter can only be accessed via the DRP interface. Because the DRP only outputs 16 bits of data per operation, two DRP transactions must be completed to read out the complete 32-bit value. *To properly read out the error counter, the user must read out the lower 16 bits first, followed by the upper 16 bits.*<br>Channel 0:<br>Lower 16 bits DRP address: `0x0484`<br>Upper 16 bits DRP address: `0x0483`<br>Channel 1:<br>Lower 16 bits DRP address: `0x0684`<br>Upper 16 bits DRP address: `0x0683`<br>***Note***: This is a read-only attribute. |

*Table 62:* **Pattern Checker DRP Read-Only Registers** *(cont'd)*

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RX_PRBSCNT | 48-bit Binary | PRBS sample counter. This counter can be reset by asserting CH[0/1]_RXPRBSCSCNTRST. The counter increments by the actual number of samples received. Counting begins after RXPRBSLOCKED is asserted High. The counter saturates at `48'hFFFFFFFFFFFF`. To get the total bits checked by the RX PRBS pattern checker, CH[0/1]_RX_PRBSCNT[47:0] needs to be multiplied by the RX internal datapath mode (RX_INT_DATA_WIDTH) used. This sample counter can only be accessed via the DRP interface. Because the DRP only outputs 16 bits of data per operation, three DRP transactions must be completed to read out the complete 48-bit value. *To properly read out the sample counter, the user must read out the lower 16 bits first, followed by the middle 16 bits, and lastly the upper 16 bits.*<br>Channel 0:<br>Lower 16 bits DRP address: `0x0480`<br>Middle 16 bits DRP address: `0x0481`<br>Upper 16 bits DRP address: `0x0482`<br>Channel 1:<br>Lower 16 bits DRP address: `0x0680`<br>Middle 16 bits DRP address: `0x0681`<br>Upper 16 bits DRP address: `0x0682`<br>***Note*:** This is a read-only attribute. |

# Using RX Pattern Checker

The GTM RX pattern checker works for all supported data widths. However, 80-bit or 160-bit RX fabric data width in PAM4 mode requires additional steps. Other data widths do not require any additional steps.

### Enable RX Pattern Checker for 80-bit or 160-bit Data Widths

1. Using the DRP interface, write the following values to CH[0/1]_RX_PCS_CFG0[4:0] (address 0x080 for CH0, 0x280 for CH1):
   a. CH[0/1]_RX_PCS_CFG0[4:0] = 0x12 for 80-bit data width mode.
   b. CH[0/1]_RX_PCS_CFG0[4:0] = 0x14 for 160-bit data width mode.
2. Enable the PRBS checker by setting CH[0/1]_RXPRBSPTN to the required value for the desired pattern.

RX PRBS pattern checker is enabled.

### Disable RX Pattern Checker for 80-bit or 160-bit Data Widths

1. Disable the PRBS checker by setting CH[0/1]_RXPRBSPTN[3:0] to `4'b0000`.
2. Using the DRP interface, write the following values to CH[0/1]_RX_PCS_CFG0[4:0] (address 0x080 for CH0, 0x280 for CH1):
   a. CH[0/1]_RX_PCS_CFG0[4:0] = 0x09 for 80-bit data width mode.
   b. CH[0/1]_RX_PCS_CFG0[4:0] = 0x0B for 160-bit data width mode.
3. Reset the RX through the GT Wizard IP. For more information on the GT Wizard IP signals refer to the *Virtex UltraScale+ FPGAs GTM Transceivers Wizard LogiCORE IP Product Guide* (PG315)
4. Wait for CH[0/1]_RXRESETDONE to toggle High.

RX PRBS pattern checker is disabled.

### Calculating Bit Error Rate (BER)

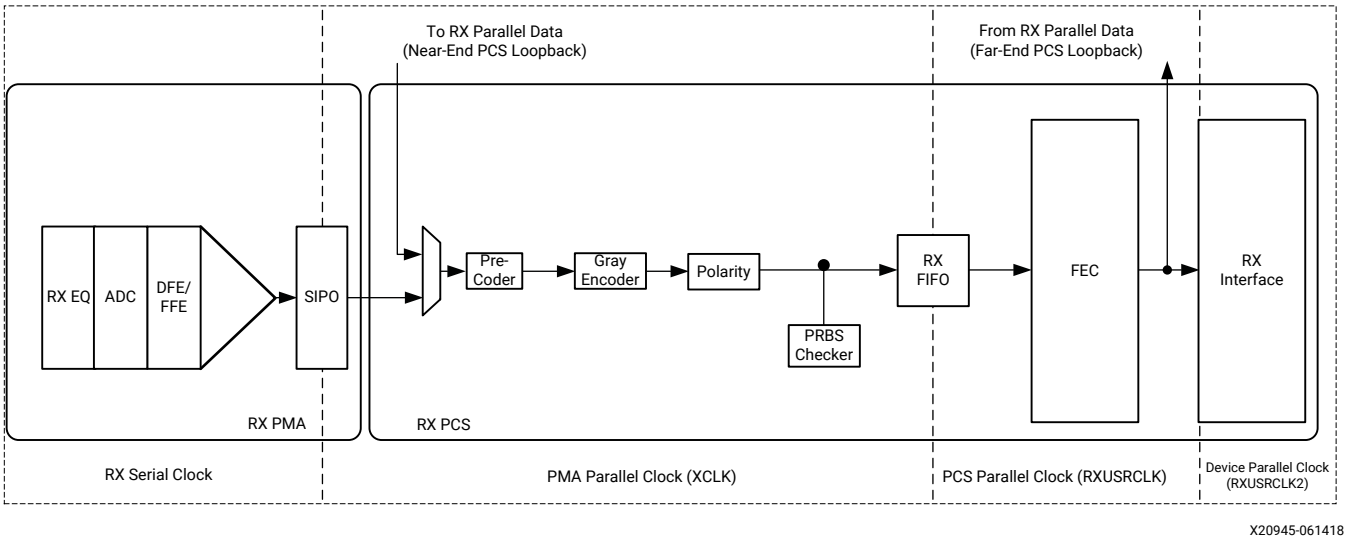
The GTM RX pattern checker can be used to calculate the BER (pre-FEC) and evaluate the performance of the high-speed link. To calculate the BER the user must follow the sequence below:

1. Reset RX PRBS sample counter and bit error counter by toggling the CH[0/1]_RXPRBSCSCNTRST port.
2. Check that the PRBS checker has locked to the incoming data by reading the CH[0/1]_RXPRBSLOCKED port.
3. Read the bit error counter through the read-only DRP attribute CH[0/1]_RX_PRBSERR[31:0].
   a. Read the lower 16-bits of the bit error counter. The bit error and sample counters are automatically latched into their read only registers in this operation.
   b. Read the upper 16-bit of the bit error counter.
4. Read the sample counter through the read-only DRP attribute CH[0/1]_RX_PRBSCNT[47:0].
   a. Read the lower 16 bits of the sample.
   b. Read the middle 16 bits of the sample counter.
   c. Read the upper 16 bits of the sample.
5. Calculate the total bits received by multiplying CH[0/1]_RX_PRBSCNT[47:0] by the RX internal datapath mode used. Check the RX internal datapath mode by reading RX_INT_DATA_WIDTH. For example, if RX_INT_DATA_WIDTH = 0x1, the multiplication factor will be 80 (see Table 72).
6. Calculate the BER by dividing the bit errors (step 3) by the total bits received (calculated in step 5). If CH[0/1]_RX_PRBSERR[31:0] = 0x0, the link is error free.

# RX Buffer

The GTM transceiver RX datapath has two internal parallel clock domains used in the PCS: the interface with PMA parallel clock domain (XCLK), and the PCS internal clock domain (RXUSRCLK). To receive data, the RX buffer provides data width conversion between these clock domains when necessary, depending on the operating data width and encoding mode. The following figure shows the RX datapath clock domains.

*Figure 45:* **RX Clock Domains**




X20945-061418

The GTM receiver includes an RX FIFO to support data width conversion when data crosses from the XCLK to RXUSRCLK domain, and the table below shows the possible scenarios. The buffer does not tolerate ppm differences and only provides phase compensation between the two clocks. The RX buffer inside the GTM transceiver must always be used. Buffer bypass is not allowed.

*Table 63:* **RX FIFO Data Width Conversion Scenarios**

| PMA Parallel Clock (XCLK) Domain Data Width | PCS Parallel Clock (RXUSRCLK) Domain Data Width | FEC Support |
|---|---|---|
| 64-bit | 64-bit | No |
| 128-bit | 80-bit | Yes |
| 128-bit | 128-bit | No |

## Ports and attributes

The following table defines the RX buffer ports.

*Table 64:* **RX Buffer Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_RXBUFSTATUS[2:0] | Out | RXUSRCLK | RX buffer status:<br>Bit[2]: Reserved.<br>Bit[1]: FIFO overflow status. A value of 1 indicates FIFO overflow.<br>Bit[0]: FIFO underflow status. A value of 1 indicates FIFO underflow. |

The following table defines the RX buffer attributes.

*Table 65:* **RX Buffer Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RX_PCS_CFG0 | 16-bit Binary | Reserved. Use the recommended value from the Wizard. |
| CH[0/1]_RXFIFO_UNDERFLOW | 1-bit Binary | A value of 1 indicates FIFO underflow.<br>For channel 0, bit[8] of DRP Address `0x48B`.<br>For channel 1, bit[8] of DRP Address `0x68B`.<br>***Note*:** This is a read-only attribute. |
| CH[0/1]_RXFIFO_OVERFLOW | 1-bit Binary | A value of 1 indicates FIFO overflow.<br>For channel 0, bit[9] of DRP Address `0x48B`.<br>For channel 1, bit[9] of DRP Address `0x68B`.<br>***Note*:** This is a read-only attribute. |

# RX FEC

The Integrated KP4 Reed-Solomon Forward Error Correction (RS-FEC) provides a robust multi-bit error detection/correction algorithm that protects up to 2 x 58 Gb/s or 1x116 Gb/s electrical and optical links. This section describes the operation of the Integrated KP4 RS-FEC within the UltraScale+™ device GTM transceivers.

KP4 FEC is based on the RS(544,514) code, which encodes message blocks of 5140 bits to produce codewords of 5440 bits. For a detailed description of the RS-FEC sublayer in Ethernet, including the definition of the KP4 FEC code, refer to clause 91 of the IEEE Standard for Ethernet (IEEE Std 802.3-2015). The same FEC code is used in other standards such as OTN FlexO and Interlaken.

The Integrated KP4 RS-FEC for each GTM dual is composed of two logical slices for each channel. These can operate as two independent RS-FEC processing units at up to 58 Gb/s each, or as one unified unit at up to 116 Gb/s. When operating at up to 116 Gb/s, data is transmitted and received over four virtual FEC lanes as described in IEEE 802.3-2015 clause 91. When operating as 2 x 58 Gb/s, data can be transmitted and received over two virtual FEC lanes per 58 Gb/s channel, or as a raw data stream (one virtual lane) with optional PN scrambling for backplane operations. The general principle of operation of the FEC is the same whichever mode is chosen.

When the RS-FEC is enabled in the receive direction, data received by the GTM PMA and PCS is passed to the Integrated KP4 RS-FEC. The RS-FEC then aligns and deskews the input lanes according to the configured mode to reconstruct the transmitted RS codeword stream. The RS-FEC then performs RS decoding according to the 802.3bj clause 91 specification. The decoded output data from the Integrated KP4 RS-FEC is then presented on the output of the GTM transceiver to the fabric.

The Integrated KP4 RS-FEC does not natively perform transcoding, alignment marker removal, alignment marker mapping, or alignment marker insertion operations in the receive directions. To support protocols such as 100G Ethernet (IEEE 802.3 clause 91) and 50G Ethernet (IEEE 802.3 clause 134) which require 257b transcoding and alignment marker processing, the GTM Wizard IP can optionally include soft logic blocks for these functions.

## Ports and Attributes

The following table shows the RX FEC-related ports for the GTM dual.

*Table 66:* **RX FEC Ports**

| Ports | Dir | Clock Domain | Description |
|---|---|---|---|
| CH[0/1]_RXFECRESET | In | Async | Component reset port for RX FEC. |
| CH[0/1]_RXDATA[159:0] | Out | CH[0/1]_RXUSRCLK2 | Output RX data. Must use 160-bit interface when FEC is enabled. |
| CH[0/1]_RXDATASTART | Out | CH[0/1]_RXUSRCLK2 | Start of codeword. |
| CH[0/1]_RXDATAISAM | Out | CH[0/1]_RXUSRCLK2 | Alignment marker flag. |
| CH[0/1]_RXDATAFLAG[3:0] | Out | CH[0/1]_RXUSRCLK2 | Codeword status flags:<br>Bit 0: When High, indicates the codeword input to the decoder has no errors.<br>Bit 1: When High, indicates the codeword input to the decoder has errors.<br>Bit 2: When High, indicates the codeword output from the decoder has no errors.<br>Bit 3: When High, indicates the codeword output from the decoder has errors. |
| FECCTRLRX0BITSLIPFS | In | CH0_RXUSRCLK2 | Bitslip control for slice 0. |
| FECCTRLRX1BITSLIPFS | In | CH1_RXUSRCLK2 | Bitslip control for slice 1. |
| FECRX0ALIGNED | Out | CH0_RXUSRCLK2 | Slice 0 alignment status. |

*Table 66:* **RX FEC Ports** *(cont'd)*

| Ports | Dir | Clock Domain | Description |
|---|---|---|---|
| FECRX1ALIGNED | Out | CH1_RXUSRCLK2 | Slice 1 alignment status. |
| FECRX0CWINC | Out | CH0_RXUSRCLK2 | Slice 0 codeword count increment. |
| FECRX1CWINC | Out | CH1_RXUSRCLK2 | Slice 1 codeword count increment. |
| FECRX0CORRCWINC | Out | CH0_RXUSRCLK2 | Slice 0 corrected codeword count increment. |
| FECRX1CORRCWINC | Out | CH1_RXUSRCLK2 | Slice 1 corrected codeword count increment. |
| FECTRXLN0LOCK | Out | CH0_RXUSRCLK2 | Lane 0 lock status. |
| FECTRXLN1LOCK | Out | CH0_RXUSRCLK2 | Lane 1 lock status. |
| FECTRXLN2LOCK | Out | CH1_RXUSRCLK2 | Lane 2 lock status. |
| FECTRXLN3LOCK | Out | CH1_RXUSRCLK2 | Lane 3 lock status. |
| FECRXLN0BITERR0TO1INC[7:0] | Out | CH0_RXUSRCLK2 | Lane 0 bit error count increment (0 corrected to 1). |
| FECRXLN1BITERR0TO1INC[7:0] | Out | CH0_RXUSRCLK2 | Lane 1 bit error count increment (0 corrected to 1). |
| FECRXLN2BITERR0TO1INC[7:0] | Out | CH1_RXUSRCLK2 | Lane 2 bit error count increment (0 corrected to 1). |
| FECRXLN3BITERR0TO1INC[7:0] | Out | CH1_RXUSRCLK2 | Lane 3 bit error count increment (0 corrected to 1). |
| FECRXLN0BITERR1TO0INC[7:0] | Out | CH0_RXUSRCLK2 | Lane 0 bit error count increment (1 corrected to 0). |
| FECRXLN1BITERR1TO0INC[7:0] | Out | CH0_RXUSRCLK2 | Lane 1 bit error count increment (1 corrected to 0). |
| FECRXLN2BITERR1TO0INC[7:0] | Out | CH1_RXUSRCLK2 | Lane 2 bit error count increment (1 corrected to 0). |
| FECRXLN3BITERR1TO0INC[7:0] | Out | CH1_RXUSRCLK2 | Lane 3 bit error count increment (1 corrected to 0). |
| FECRXLN0ERRCNTINC[3:0] | Out | CH0_RXUSRCLK2 | Lane 0 symbol error count increment. |
| FECRXLN1ERRCNTINC[3:0] | Out | CH0_RXUSRCLK2 | Lane 1 symbol error count increment. |
| FECRXLN2ERRCNTINC[3:0] | Out | CH1_RXUSRCLK2 | Lane 2 symbol error count increment. |
| FECRXLN3ERRCNTINC[3:0] | Out | CH1_RXUSRCLK2 | Lane 3 symbol error count increment. |
| FECRXLN0MAPPING[1:0] | Out | CH0_RXUSRCLK2 | Logical FEC lane mapped to physical lane 0. |
| FECRXLN1MAPPING[1:0] | Out | CH0_RXUSRCLK2 | Logical FEC lane mapped to physical lane 1. |
| FECRXLN2MAPPING[1:0] | Out | CH1_RXUSRCLK2 | Logical FEC lane mapped to physical lane 2. |
| FECRXLN3MAPPING[1:0] | Out | CH1_RXUSRCLK2 | Logical FEC lane mapped to physical lane 3. |
| FECRXLN0DLY[14:0] | Out | CH0_RXUSRCLK2 | Lane 0 alignment delay. |
| FECRXLN1DLY[14:0] | Out | CH0_RXUSRCLK2 | Lane 1 alignment delay. |
| FECRXLN2DLY[14:0] | Out | CH1_RXUSRCLK2 | Lane 2 alignment delay. |
| FECRXLN3DLY[14:0] | Out | CH1_RXUSRCLK2 | Lane 3 alignment delay. |

The receive portion of the Integrated KP4 RS-FEC operates internally in the CH0_RXUSRCLK and CH1_RXUSRCLK domains. Data output on CH0_RXDATA is clocked on the rising edge of CH0_RXUSRCLK2, and data output on CH1_RXDATA is clocked on the rising edge of CH1_RXUSRCLK2, just as when the FEC is not enabled.

When configured in 100G mode (combined slice 0 and slice 1 operation), CH0_RXUSRCLK and CH1_RXUSRCLK must be driven from the same source with low skew. The same applies to CH0_RXUSRCLK2 and CH1_RXUSRCLK2.

When the Integrated KP4 RS-FEC is configured in 2 x 50G mode, the overall statistics for slice 0's decoder are reported on FECRX0*, with slice 0 lane 0's statistics on FECRXLN0*, and slice 0 lane 1's statistics on FECRXLN1*. The overall statistics for slice 1's decoder are reported on FECRX1*, with slice 0 lane 0's statistics on FECRXLN2*, and slice 0 lane 1's statistics on FECRXLN3*.

When the Integrated KP4 RS-FEC is configured in 100G mode, the overall statistics for the RS decoder are reported on FECRX0*. The FECRX1* ports should be ignored in this mode. The statistics for the four lanes are reported on FECRXLN0* through FECRXLN3*.

The following table shows the RX FEC-related attributes for the GTM dual.

*Table 67:* **RX FEC Attributes**

| Attribute | Type | Description |
|---|---|---|
| FEC_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| FEC_RX0_MODE | [11:8] | Operation mode for FEC RX slice 0:<br>`4'b0000`: FEC is disabled for this channel.<br>`4'b0001`: 50G KP4 FEC, 50GAUI-1 format.<br>`4'b0010`: 100G KP4 FEC, 100GAUI-2 format.<br>`4'b0101`: 50G raw KP4 FEC without scrambling.<br>`4'b1101`: 50G raw KP4 FEC with scrambling.<br>Others: Invalid. |
| FEC_RX1_MODE | [15:12] | Operation mode for FEC RX slice 1:<br>`4'b0000`: FEC is disabled for this channel.<br>`4'b0001`: 50G KP4 FEC, 50GAUI-1 format.<br>4'b0010: 100G KP4 FEC, 100GAUI-2 format.<br>`4'b0101`: 50G raw KP4 FEC without scrambling.<br>`4'b1101`: 50G raw KP4 FEC with scrambling.<br>Others: Invalid. |
| FEC_CFG3 | 16-bit | Reserved. |

*Table 67:* **RX FEC Attributes** *(cont'd)*

| Attribute | Type | Description |
|---|---|---|
| **Bit Name** | **Address** | **Description** |
| FEC_RX0_BYPASS_CORRECTION | [0] | FEC RX slice 0 error correction select:<br>`1'b0`: Error correct enabled.<br>`1'b1`: Error correct disabled. |
| FEC_RX1_BYPASS_CORRECTION | [1] | FEC RX slice 1 error correction select:<br>`1'b0`: Error correct enabled.<br>`1'b1`: Error correct disabled. |
| FEC_RX0_BYPASS_INDICATION | [2] | FEC RX slice 0 error indicator enable:<br>`1'b0`: Error indicator enabled.<br>`1'b1`: Error indicator disabled. |
| FEC_RX1_BYPASS_INDICATION | [3] | FEC RX slice 1 error indicator enable:<br>`1'b0`: Error indicator enabled.<br>`1'b1`: Error indicator disabled. |
| FEC_CFG1[15:0] (FEC_RX0_VL_LENGTH[15:0])<br>FEC_CFG3[4] (FEC_RX0_VL_LENGTH[16]) | 17-bit Binary | Frame length (in 66-bit blocks per PCS lane) for FEC RX slice 0. Ignored if FEC RX slice 0 is configured in raw FEC mode. |
| FEC_CFG2[15:0] (FEC_RX1_VL_LENGTH[15:0])<br>FEC_CFG3[5] (FEC_RX1_VL_LENGTH[16]) | 17-bit Binary | Frame length (in 66-bit blocks per PCS lane) for FEC RX slice 1. Ignored if FEC RX slice 1 is configured in raw FEC mode. |
| FEC_CFG4[15:0] (FEC_RX_VL_MARKER_ID0[15:0])<br>FEC_CFG5[15:0] (FEC_RX_VL_MARKER_ID0[31:16])<br>FEC_CFG6[15:0] (FEC_RX_VL_MARKER_ID0[47:32])<br>FEC_CFG7[15:0] (FEC_RX_VL_MARKER_ID0[63:48]) | 64-bit Binary | Alignment marker 0 (common), for 50GE operation on FEC RX slice 0, and for 100GE, FlexO, and Interlaken operation on FEC RX slices 0–1. |
| FEC_CFG8[15:0] (FEC_RX_VL_MARKER_ID1[15:0])<br>FEC_CFG9[15:0] (FEC_RX_VL_MARKER_ID1[31:16])<br>FEC_CFG10[15:0]<br>(FEC_RX_VL_MARKER_ID1[47:32])<br>FEC_CFG11[15:0]<br>(FEC_RX_VL_MARKER_ID1[63:48]) | 64-bit Binary | Alignment marker 0 (common) for 50GE operation on FEC RX slice 1. Ignored if FEC RX slice 0 is configured in 100G mode. |
| FEC_CFG12[15:0] (FEC_RX_VL_MARKER_ID4[15:0])<br>FEC_CFG13[15:0]<br>(FEC_RX_VL_MARKER_ID4[31:16])<br>FEC_CFG14[15:0]<br>(FEC_RX_VL_MARKER_ID4[47:32])<br>FEC_CFG15[15:0]<br>(FEC_RX_VL_MARKER_ID4[63:48]) | 64-bit Binary | Alignment marker 4 (lane-specific) for 100GE, FlexO, and Interlaken operation on FEC RX slice 0–1. Alignment marker 2 (lane-specific) for 50GE operation on FEC RX slice 0. |
| FEC_CFG16[15:0] (FEC_RX_VL_MARKER_ID5[15:0])<br>FEC_CFG17[15:0]<br>(FEC_RX_VL_MARKER_ID5[31:16])<br>FEC_CFG18[15:0]<br>(FEC_RX_VL_MARKER_ID5[47:32])<br>FEC_CFG19[15:0]<br>(FEC_RX_VL_MARKER_ID5[63:48]) | 64-bit Binary | Alignment marker 5 (lane-specific) for 100GE, FlexO, and Interlaken operation on FEC RX slice 0–1. |

# RS-FEC Sub-Modes

There are three sub-modes of RS-FEC decoder operation defined by the IEEE 802.3bj specification. The primary purpose of the different sub-modes is to allow latency to be reduced in cases where the line BER is low enough that full error correction is not required. The following table lists the valid RS-FEC operating sub-modes. When configured in 2x50G FEC mode, the sub-mode can be set independently for each slice.

*Table 68:* **Valid RS-FEC Operating Sub-Modes**

| Control Signal | Sub-Modes | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| FEC_RX*n*_BYPASS_INDICATION | 0 | 0 | 1 |
| FEC_RX*n*_BYPASS_CORRECTION | 0 | 1 | 0 |

### Sub-mode 1: Full Operation

In this sub-mode, the RS-FEC engine detects and corrects errors. If a codeword with up to 15 symbol errors is received, the errors are corrected. If a codeword with 16 or more symbol errors is received, the erroneous data is flagged soon after the start of data output.

### Sub-mode 2: Error Indication, No Error Correction

In this sub-mode, the RS-FEC engine detects errors but does not correct them. If a codeword with one or more errors is received, no error correction is attempted, but the erroneous data is flagged soon after the start of data output.

### Sub-mode 3: Error Correction, No Error Indication

In this sub-mode, the RS-FEC engine detects and corrects errors. If a codeword with up to 15 symbol errors is received, the errors are corrected. If a codeword with 16 or more symbol errors is received, no correction is possible, and the erroneous data is not flagged until after it has been output.

# Usage Model

When FEC is enabled in 50G mode, bit 0 of each RXDATA bus is the first bit received in time, and bit 159 is the last bit received in time. The RXDATASTART signal is driven High whenever the RXDATA for the associated slice contains the first 160 bits of a codeword. Output codewords are always aligned so that bit 0 of a codeword is on bit 0 of the RXDATA bus.

When FEC is enabled in 100G mode, bit 0 of CH0_RXDATA is the first bit received in time, and bit 159 of CH1_RXDATA is the last bit received in time. The CH0_RXDATASTART signal is driven High whenever the RXDATA buses contain the first 320 bits of a codeword. Output codewords are always aligned so that bit 0 of a codeword is on bit 0 of CH0_RXDATA. CH1_RXDATASTART should be ignored in 100G mode.

The RXDATAISAM signal is driven High alongside RXDATASTART whenever the codeword received on the associated slice contains an alignment marker. This signal is not active when the FEC is configured in 50G raw mode. In 100G mode, CH1_RXDATAISAM should be ignored.

In all modes, codewords are 5440 bits in length. The final 300 bits of output data (bits 5140 to 5439) is the FEC parity as received and possibly corrected by the RS decoder. In most applications, the parity bits can be discarded.

### 50G Ethernet

Up to two channels of 50G Ethernet with KP4 FEC can be implemented as per *IEEE Draft Standard for Ethernet Amendment: Media Access Control Parameters for 50 Gb/s and Physical Layers and Management Parameters for 50 Gb/s, 100 Gb/s, and 200 Gb/s Operation* (IEEE Std 802.3cd Clause 134). Transcoding should be enabled in the GTM Wizard IP for this mode. The nominal pre-FEC PCS data rate is 51.5625 Gb/s, and the nominal post-FEC line rate is 53.125 Gb/s.

### 100G Ethernet

One channel of 100G Ethernet with KP4 FEC can be implemented as per IEEE Std 802.3-2015 Clause 91. Transcoding should be enabled in the GTM Wizard IP for this mode. The nominal aggregate pre-FEC PCS data rate is 103.125 Gb/s and the nominal aggregate post-FEC line rate is 106.25 Gb/s.

### 100G OTN FlexO

One channel of 100G OTN FlexO with a KP4 FEC can be implemented as per ITU-T G.709.1, *Flexible OTN Short-Reach Interface*. Transcoding should be disabled in the GTM Wizard IP for this mode. The nominal aggregate post-FEC line rate is 111.81 Gb/s.

### 100G Interlaken

One channel of 100G Interlaken with KP4 FEC can be implemented as per the *Interlaken Reed-Solomon Forward Error Correction Extension Protocol Definition*. Transcoding should be disabled in the GTM Wizard IP for this mode. Line rates up to 58 Gb/s are possible.

### *Proprietary Backplane Protocols with FEC up to 58 Gb/s*

When FEC is enabled in 50G raw mode (with or without scrambling), driving FECCTRLRXnBITSLIPFS High causes the FEC decoder in slice n to move the starting position of the decoded codeword to the next bit position. By repeatedly slipping the serial data in this way until FEC decoding is successful, alignment to the incoming codeword stream can be achieved without the need for alignment markers to be inserted in the data.

When the FEC is enabled in any mode other than the 50G raw modes, these signals are ignored.

# RX Interface

The RX interface is the gateway to the RX datapath of the GTM transceiver. Applications receive data through the GTM transceiver by reading data to the RXDATA port on the positive edge of RXUSRCLK2. Port widths can be 64 and 128 bits for NRZ mode, or 80, 128, 160, and 256 bits for PAM4 mode. The rate of the parallel clock (RXUSRCLK2) at the interface is determined by the RX line rate and the width of the RXDATA port. A second parallel clock (RXUSRCLK) must be provided for the internal PCS logic in the receiver. This section shows how to drive the parallel clocks and explains the constraints on those clocks for correct operation.

## Interface Width Configuration

The GTM transceiver contains a 64-bit internal datapath in NRZ mode, and an 80-bit and 128-bit internal datapath in PAM4 mode that is configurable by setting the RX_INT_DATA_WIDTH attribute. When the FEC is enabled, only the 80-bit internal datapath can be used. The interface width is configurable by setting the RX_DATA_WIDTH attribute. In NRZ mode, RX_DATA_WIDTH can be configured to 64 or 128 bits. In PAM4 mode, RX_DATA_WIDTH can be configured to 80, 128, 160, or 256 bits. When the FEC is enabled, only the 80-bit or 160-bit RX internal data width can be selected.

The following table shows how the interface width for RX datapath is selected.

*Table 69:* **RX Interface Datapath Configuration**

| Encoding | FEC Allowed? | RX_DATA_WIDTH Encoding | RX Data Width Selection | RX_INT_DATA_WIDTH Encoding | RX Internal Datapath Selection |
|---|---|---|---|---|---|
| NRZ | No | 0 | 64 | 0 | 64 |
| | No | 2 | 128 | 0 | 64 |
| PAM4 | Yes | 1 | 80 | 1 | 80 |
| | Yes | 3 | 160 | 1 | 80 |
| | No | 2 | 128 | 2 | 128 |
| | No | 4 | 256 | 2 | 128 |

The following figure shows the RX data received.

*Figure 46:* **RX Data Received**

| | <<<Data Reception Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Data Received | RXDATA[31:24] | | | | | | | | RXDATA[23:16] | | | | | | | | RXDATA[15:8] | | | | | | | | RXDATA[7:0] | | | | | | | |

| | <<<Data Reception Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| Data Received | RXDATA[63:56] | | | | | | | | RXDATA[55:48] | | | | | | | | RXDATA[47:40] | | | | | | | | RXDATA[39:32] | | | | | | | |

| | <<<Data Reception Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 96 | 95 | 94 | 93 | 92 | 91 | 90 | 89 | 88 | 87 | 86 | 85 | 84 | 83 | 82 | 81 | 80 | 79 | 78 | 77 | 76 | 75 | 74 | 73 | 72 | 71 | 70 | 69 | 68 | 67 | 66 | 65 |
| Data Received | RXDATA[96:89] | | | | | | | | RXDATA[88:81] | | | | | | | | RXDATA[80:73] | | | | | | | | RXDATA[72:65] | | | | | | | |

| | <<<Data Reception Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 127 | 126 | 125 | 124 | 123 | 122 | 121 | 120 | 119 | 118 | 117 | 116 | 115 | 114 | 113 | 112 | 111 | 110 | 109 | 108 | 107 | 106 | 105 | 104 | 103 | 102 | 101 | 100 | 99 | 98 | 97 | 96 |
| Data Received | RXDATA[127:120] | | | | | | | | RXDATA[119:112] | | | | | | | | RXDATA[111:104] | | | | | | | | RXDATA[103:96] | | | | | | | |

| | <<<Data Reception Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 159 | 158 | 157 | 156 | 155 | 154 | 153 | 152 | 151 | 150 | 149 | 148 | 147 | 146 | 145 | 144 | 143 | 142 | 141 | 140 | 139 | 138 | 137 | 136 | 135 | 134 | 133 | 132 | 131 | 130 | 129 | 128 |
| Data Received | RXDATA[159:152] | | | | | | | | RXDATA[151:144] | | | | | | | | RXDATA[143:136] | | | | | | | | RXDATA[135:128] | | | | | | | |

| | <<<Data Reception Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 191 | 190 | 189 | 188 | 187 | 186 | 185 | 184 | 183 | 182 | 181 | 180 | 179 | 178 | 177 | 176 | 175 | 174 | 173 | 172 | 171 | 170 | 169 | 168 | 167 | 166 | 165 | 164 | 163 | 162 | 161 | 160 |
| Data Received | RXDATA[191:184] | | | | | | | | RXDATA[183:176] | | | | | | | | RXDATA[175:168] | | | | | | | | RXDATA[167:160] | | | | | | | |

| | <<<Data Reception Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 223 | 222 | 221 | 220 | 219 | 218 | 217 | 216 | 215 | 214 | 213 | 212 | 211 | 210 | 209 | 208 | 207 | 206 | 205 | 204 | 203 | 202 | 201 | 200 | 199 | 198 | 197 | 196 | 195 | 194 | 193 | 192 |
| Data Received | RXDATA[223:216] | | | | | | | | RXDATA[215:208] | | | | | | | | RXDATA[207:200] | | | | | | | | RXDATA[199:192] | | | | | | | |

| | <<<Data Reception Order is Right to Left (LSB to MSB)<<< | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 255 | 254 | 253 | 252 | 251 | 250 | 249 | 248 | 247 | 246 | 245 | 244 | 243 | 242 | 241 | 240 | 239 | 238 | 237 | 236 | 235 | 234 | 233 | 232 | 231 | 230 | 229 | 228 | 227 | 226 | 225 | 224 |
| Data Received | RXDATA[255:248] | | | | | | | | RXDATA[247:240] | | | | | | | | RXDATA[239:232] | | | | | | | | RXDATA[231:224] | | | | | | | |

## *RXUSRCLK and RXUSRCLK2 Generation*

The RX interface includes two parallel clocks: RXUSRCLK and RXUSRCLK2. RXUSRCLK is the internal clock for the PCS logic in the GTM transmitter. The required rate for RXUSRCLK depends on the internal datapath width of the GTM_DUAL primitive and the RX line rate of the GTM transmitter. The following equation shows how to calculate the required rate for RXUSRCLK for all cases.

$$RXUSRCLK\quad Rate = \frac{Line\quad Rate}{Internal\quad Datapath\quad Width}$$

RXUSRCLK2 is the main synchronization clock for all signals into the RX side of the GTM transceiver. Most signals into the RX side of the GTM transceiver are sampled on the positive edge of RXUSRCLK2. RXUSRCLK2 and RXUSRCLK have a fixed-rate relationship based on the RX_DATA_WIDTH and RX_INT_DATA_WIDTH settings. The following table shows the relationship between RXUSRCLK2 and RXUSRCLK per the RX_DATA_WIDTH and RX_INT_DATA_WIDTH values.

*Table 70:* **RXUSRCLK2 Frequency Relationship to RXUSRCLK**

| Encoding | RX Data Width | RX Internal Datapath | RXUSRCLK2 Frequency |
|---|---|---|---|
| NRZ | 64 | 64 | $F_{RXUSRCLK2} = F_{RXUSRCLK}$ |
| | 128 | 64 | $F_{RXUSRCLK2} = F_{RXUSRCLK}/2$ |
| PAM4 | 80 | 80 | $F_{RXUSRCLK2} = F_{RXUSRCLK}$ |
| | 160 | 80 | $F_{RXUSRCLK2} = F_{RXUSRCLK}/2$ |
| | 128 | 128 | $F_{RXUSRCLK2} = F_{RXUSRCLK}$ |
| | 256 | 128 | $F_{RXUSRCLK2} = F_{RXUSRCLK}/2$ |

These rules about the relationships between clocks must be observed for RXUSRCLK and RXUSRCLK2:

- RXUSRCLK and RXUSRCLK2 must be positive-edge aligned, with as little skew as possible between them. As a result, low-skew clock resources (BUFG_GTs) must be used to drive RXUSRCLK and RXUSRCLK2.
- If the channel is configured so that the same oscillator drives the reference clock for the transmitter and the receiver, TXPROGDIVCLK can be used to drive RXUSRCLK and RXUSRCLK2 in the same way that they are used to drive TXUSRCLK and TXUSRCLK2.
- If separate oscillators are driving the reference clocks for the transmitter and receiver on the channel, RXUSRCLK and RXUSRCLK2 must be driven by RXPROGDIVCLK.

## *Ports and Attributes*

The following table defines the TX interface ports.

*Table 71:* **RX Interface Ports**

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| RXDATA[255:0] | Out | RXUSRCLK2 | The bus for receiving data. The width of this port is equal to RX data width selection. RX data width selection:<br>64: RXDATA[63:0]<br>80: RXDATA[79:0]<br>128: RXDATA[127:0]<br>160: RXDATA[159:0]<br>256: RXDATA[255:0] |

*Table 71:* **RX Interface Ports** *(cont'd)*

| Port | Dir | Clock Domain | Description |
|---|---|---|---|
| RXUSRCLK | In | Clock | This port is used to provide a clock for the internal RX PCS datapath. |
| RXUSRCLK2 | In | Clock | This port is used to synchronize the interconnect logic with the RX interface. This clock must be positive-edge aligned to RXUSRCLK. |

The following table defines the TX interface attributes.

*Table 72:* **RX Interface Attributes**

| Attribute | Type | Description |
|---|---|---|
| CH[0/1]_RX_PCS_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| RX_DATA_WIDTH | [2:0] | Sets the bit width of the RXDATA port. When FEC is enabled, RX_DATA_WIDTH must be set to 160:<br>0x0: 64-bit fabric mode.<br>0x1: 80-bit fabric mode.<br>0x2: 128-bit fabric mode.<br>0x3: 160-bit fabric mode.<br>0x4: 256-bit fabric mode. |
| RX_INT_DATA_WIDTH | [4:3] | Controls the width of the internal RX PCS datapath. 80-bit internal datapath must use with 80- or 160-bit fabric width; 128-bit internal datapath must use with 128- or 256-bit fabric width; 64-bit internal datapath must use with 64 or 128-bit fabric width:<br>0x0: 64-bit internal datapath mode.<br>0x1: 80-bit internal datapath mode.<br>0x2: 128-bit internal datapath mode. |
| GEN_RXUSRCLK | [14] | Automatically generates RXUSRCLK from RXUSRCLK2, only applicable when fabric datapath width is the same as the internal datapath width:<br>0x0: Disables automatic RXUSRCLK generation from RXUSRCLK2.<br>0x1: Enables automatic RXUSRCLK generation from RXUSRCLK2. |
| CH[0/1]_A_CH_CFG0 | 16-bit | Reserved. |
| **Bit Name** | **Address** | **Description** |
| RX_FABINT_USRCLK_FLOP | [1] | Determines if port signals are registered again in the RXUSRCLK domain after being registered in the RXUSRCLK2 domain. This attribute only applies if the RX internal datapath width is the same as the RX interface width, otherwise this attribute is ignored. Use the recommended value from the Wizard.<br>0x0: Bypass RXUSRCLK flip-flops.<br>0x1: Enable RXUSRCLK flip-flops. |

# Board Design Guidelines

This chapter presents topics related to implementing a design on a printed circuit board that uses GTM transceivers. GTM transceivers are analog circuits that require special consideration and attention when designing and implementing on a printed circuit board. Besides an understanding of the functionality of the device pins, a design that performs optimally requires attention to issues such as device interfacing, transmission line impedance and routing, power supply design filtering and distribution, component selection, and PCB layout and stackup design.

## Pin Description and Design Guidelines

The following table defines the GTM transceiver Dual pins.

*Table 73:* **GTM Transceiver Dual Pin Descriptions**

| Pins | Dir | Description |
|---|---|---|
| MGTREFCLKP<br>MGTREFCLKN | In/Out (Pad) | Configured as either reference clock input pins or as RX recovered clock output pins for the Dual. |
| MGTMRXP[1:0]<br>MGTMRXN[1:0] | In (Pad) | RXP and RXN are the differential input pairs for each of the receivers in the GTM transceiver Dual. |
| MGTMTXP[1:0]<br>MGTMTXN[1:0] | Out (Pad) | TXP and TXN are the differential output pairs for each of the transceivers in the GTM transceiver Dual. |
| MGTAVTTRCAL | In (Pad) | Bias current supply for the termination resistor calibration circuit. See Termination Resistor Calibration Circuit. |
| MGTRREF | In (Pad) | Calibration resistor input pin for the termination resistor calibration circuit. See Termination Resistor Calibration Circuit. |
| MGTAVCC | In (Pad) | MGTAVCC is the analog supply for the internal analog circuits of the GTM transceiver Dual tile. This includes the analog circuits for the PLL, transmitters, and receivers. Most packages have multiple groups of power supply connections in the package for MGTAVCC. Refer to the package pin definitions to identify the location of the power supply group for a specific GTM transceiver Dual.The nominal voltage is 0.9 VDC. |
| MGTAVTT | In (Pad) | MGTAVTT is the analog supply for the transmitter and receiver termination circuits of the GTM transceiver Dual tile. Most packages have multiple groups of power supply connections in the package for MGTAVTT. Refer to the package pin definitions to identify the location of the power supply group for a specific GTM transceiver Dual. The nominal voltage is 1.2 VDC. |

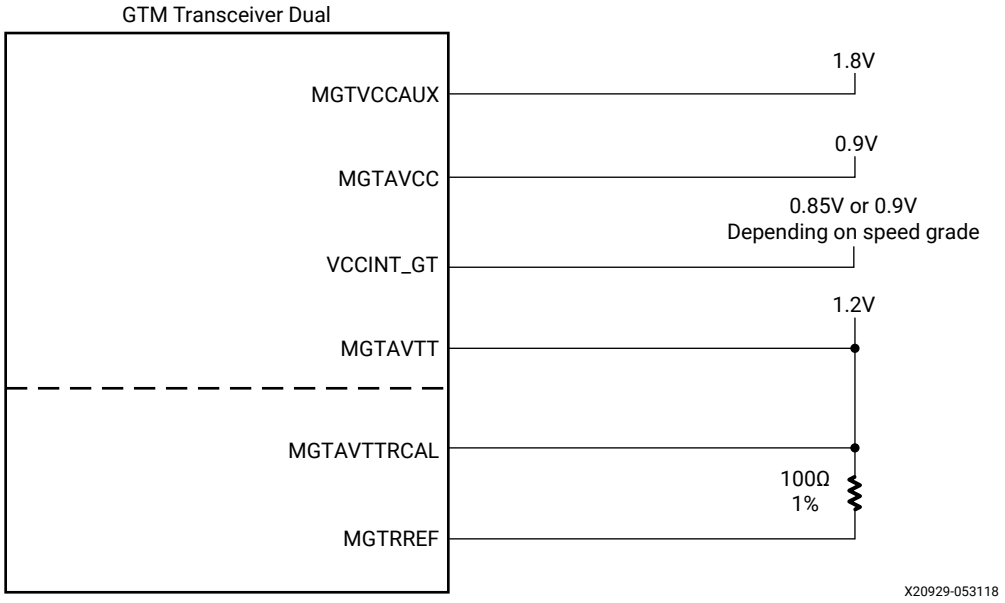
*Table 73:* **GTM Transceiver Dual Pin Descriptions** *(cont'd)*

| Pins | Dir | Description |
|---|---|---|
| MGTVCCAUX | In (Pad) | MGTVCCAUX is the auxiliary analog LCPLL voltage supply for the transceivers. Most packages have multiple groups of power supply connections in the package for MGTVCCAUX. Refer to the package pin definitions to identify in which power supply group a specific GTM transceiver Dual is located. The nominal voltage is 1.8 VDC. |
| VCCINT_GT | In (Pad) | VCCINT_GT is the supply voltage for digital circuits in GTM transceivers. Most packages have multiple groups of power supply connections for VCCINT_GT. Refer to the package pin definitions to identify the location of the power supply group for a specific GTM transceiver Dual. For all speed grades other than -2LE, VCCINT and VCCINT_GT can be connected to the same power regulation circuit. Any time VCCINT and VCCINT_GT are connected to separate voltage regulation circuits, VCCINT_GT must be within the operating voltage range prior to device configuration. The nominal voltage is 0.9 VDC or 0.85 VDC depending on the speed grade. See the UltraScale+ device data sheets (see http://www.xilinx.com/documentation) for details. |

***Note*:** In -2LE devices, VCCINT = 0.72V, but VCCINT_GT must still be maintained at 0.85V. In all other speed grades, VCCINT and VCCINT_GT are at the same voltage level.

The following figure shows the external power supply connections with the GTM transceivers.

*Figure 47:* **GTM Transceivers External Power Supply Connections**




Note relevant to the figure:

1. The voltage values are nominal. See the UltraScale+ device data sheets (see http://www.xilinx.com/documentation) for values and tolerances.

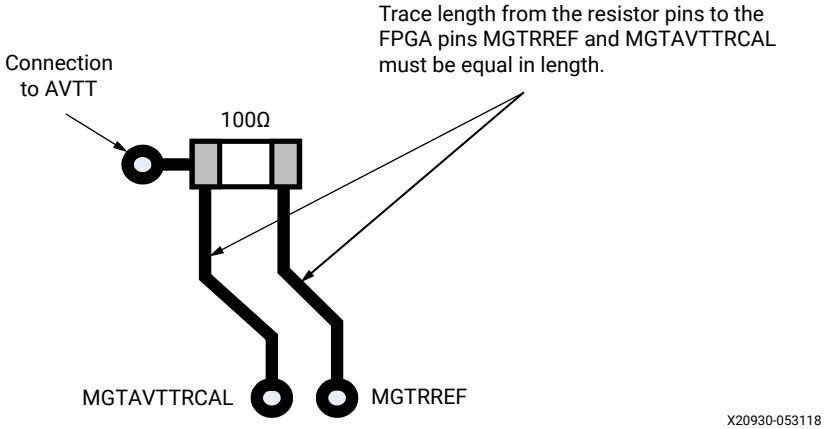
# Termination Resistor Calibration Circuit

One resistor calibration circuit (RCAL) is shared between all GTM transceiver DUAL primitives in a GTM transceiver Dual column. The MGTAVTTRCAL and MGTRREF pins connect the bias circuit power and the external calibration resistor to the RCAL circuit. The RCAL circuit performs the resistor calibration only during configuration of the UltraScale+ device. Prior to configuration, all analog supply voltages must be present and within the proper tolerance as specified in the UltraScale+ device data sheets (see http://www.xilinx.com/documentation). If an entire power supply group (PSG) is not used by any Duals, MGTAVTTRCAL and MGTRREF should be tied to ground. See Analog Power Supply Pins for more details regarding RCAL biasing recommendations when there are unused Duals.

The RCAL circuit is associated with the GTM transceiver Dual that is the RCAL master. The RCAL master performs the termination resistor calibration during configuration of the UltraScale+ device and then distributes the calibrated values to all of the GTM transceiver Duals in the column. The Dual in which the RCAL circuit is located must be powered on. For Stacked Silicon Interconnect (SSI) technology devices, each slice to be used (that contains multiple Duals) must be powered on.

Connect the MGTAVTTRCAL pin to the MGTAVTT supply and to a pin on the 100Ω precision external resistor. The other pin of the resistor is connected to the MGTRREF pin. The resistor calibration circuit provides a controlled current load to the resistor connected to the MGTRREF pin. It then senses the voltage drop across the external calibration resistor and uses that value to adjust the internal resistor calibration setting. The quality of the resistor calibration is dependent on the accuracy of the voltage measurement at the MGTAVTTRCAL and MGTRREF pins. To eliminate errors due to the voltage drop across the traces that lead from the resistor and to the UltraScale+ device pins, the trace from the MGTAVTTRCAL pin to the resistor should have the same length and geometry as the trace that connects the other pin of the resistor to the MGTRREF pin. Also, the maximum DC resistance of the PCB trace must be limited to less than 0.5Ω. (See the suggested layout in the following figure.)

*Figure 48:* **PCB Layout for the RCAL Resistor**



# Analog Power Supply Pins

The GTM transceiver Dual power supplies (MGTAVCC, MGTAVTT, VCCINT_GT, and MGTVCCAUX) have planes inside the package. For some of the packages, there are multiple planes for each analog power supply. If there is more than one PSG in the package, the power supply pin names have a suffix (such as _LN, _RN, _LS, or _RS) that identifies which pins are associated with which PSG. If all of the Duals in a PSG are not used, the associated power pins can be left unconnected or tied to GND. The rules for powering PSGs are as follows:

- Within a package PSG, if no Duals are used, the PSG can be unpowered.
- If any Duals in a PSG are used, the PSG must be powered.
- PSGs on each side (left or right) of the package are fully independent. Powering or not powering PSGs on one side of the package does not affect the PSGs on the other sides of the package.
- If a PSG does not have an RCAL master and it is powered, all the PSGs on that side (left or right) of the package must be powered.
- If a PSG with an RCAL master is unpowered, any PSGs without an RCAL master on that side of the package must also be unpowered.
- A PSG that does not have an RCAL master can be unpowered without affecting other PSGs.

For each GTM transceiver there are three analog power supplies and one digital power supply. The analog power supplies are MGTMAVCC, MGTMAVTT, and MGTVCCAUX, while the digital power supply is VCCINT_GT. In the package, the GTM analog power supplies are organized into power supply groups called PSGs. Each PSG has a power plane in the package for each of the three analog power supplies. For example, if there are two analog PSGs in a package, there would be six analog power planes in the package.

The VCCINT_GT digital power supply is organized differently from that of the analog power supplies. The VCCINT_GT power supply has a single separate power supply plane for each device with GTM transceivers. For example, if there are GTM transceivers on the right and left side of the device, there will be two VCCINT_GT digital PSGs in the package. The following table shows the PSGs for Virtex UltraScale+ devices.

*Figure 49:* **UltraScale+ Device Transceiver Power Supply Groups and RCAL Master**

| Device | Package | GTM | | | | | | | | | | | | | | | | | | | | | | | GTY | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 120 | 121 | 122 | 123 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 220 | 221 | 222 | 223 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 124 | 125 | 126 | 127 | 224 | 225 | 226 | 227 |
| xcvu27p | figd2104 | | | | | LN | LN RCL | LN | LN | | LN RCL | | | | | | | RS | RN RCL | RN | RN | RN | RN RCL | | | | | | | RS | RS RCL | RS | RS |
| | fsga2577 | | | | | LUC | LUC RCL | LUC | LUC | LN | LN RCL | LN | LN | | | | | RUC | RUC RCL | RUC | RUC | RN | RN RCL | RN | RN | LLC | LLC RCL | LLC | LLC | RLC | RLC RCL | RLC | RLC |
| xcvu29p | figd2104 | LS | LS RCL | LS | LS | LN | LN RCL | LN | LN | | LN RCL | | | | | | | RS | RN RCL | RN | RN | RN | RN RCL | | | | | | | RS | RS RCL | RS | RS |
| | fsga2577 | LS | LS RCL | LS | LS | LUC | LUC RCL | LUC | LUC | LN | LN RCL | LN | LN | RS | RS RCL | RS | RS | RUC | RUC RCL | RUC | RUC | RN | RN RCL | RN | RN | LLC | LLC RCL | LLC | LLC | RLC | RLC RCL | RLC | RLC |

# Reference Clock

This section focuses on the selection of the reference clock source or oscillator. An oscillator is characterized by:

- Frequency range
- Output voltage swing
- Jitter (deterministic, random, peak-to-peak)
- Rise and fall times
- Supply voltage and current
- Noise specification
- Duty cycle and duty-cycle tolerance
- Frequency stability

These characteristics are selection criteria when choosing an oscillator for a GTM transceiver design. Figure 50 illustrates the convention for the single-ended clock input voltage swing, peak-to-peak. This figure is provided to show the contrast to the differential clock input voltage swing calculation shown in Figure 51, as used in the GTM transceiver portion of the UltraScale+ device data sheets (see http://www.xilinx.com/documentation).

*Figure 50:* **Single-Ended Clock Input Voltage Swing, Peak-to-Peak**

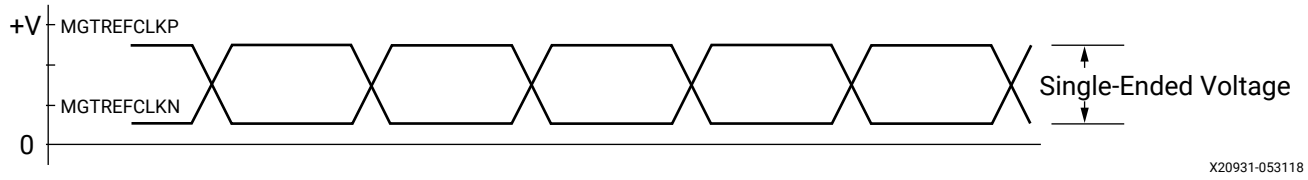



Figure 51 illustrates the differential clock input voltage swing, which is defined as MGTREFCLKP - MGTREFCLKN.

*Figure 51:* **Differential Clock Input Voltage Swing, Peak-to-Peak**

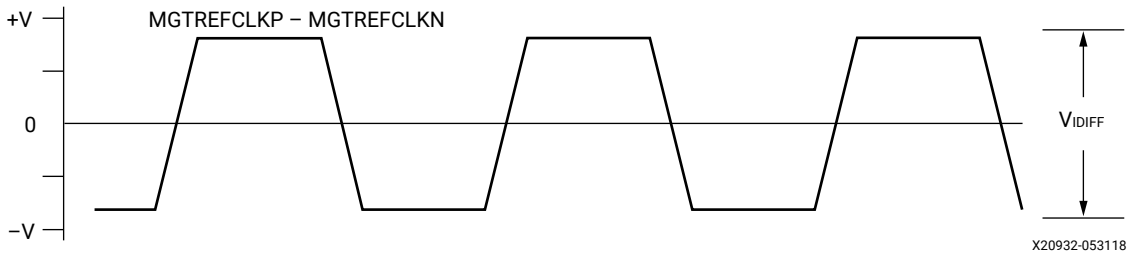
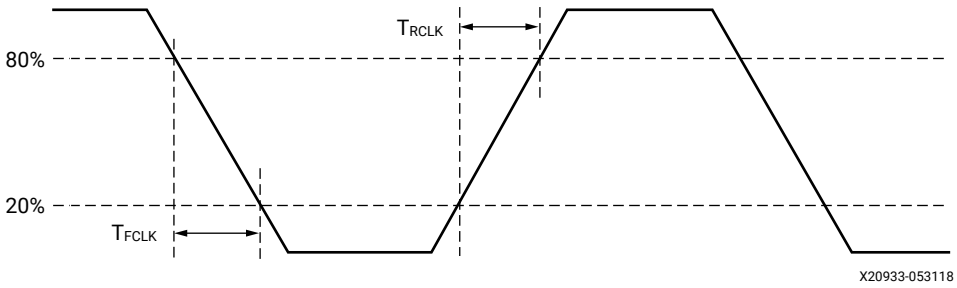


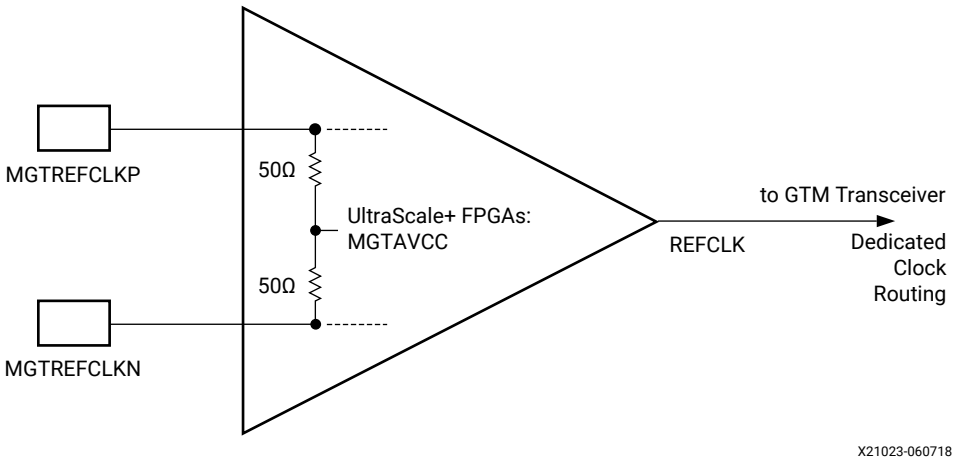

Figure 52 shows the rise and fall time convention of the reference clock.

*Figure 52:* **Rise and Fall Times**




The following figure illustrates the internal details of the IBUFDS. The dedicated differential reference clock input pair MGTREFCLKP/MGTREFCLKN is internally terminated with 100Ω differential impedance. The common mode voltage of this differential reference clock input pair is 4/5 of MGTAVCC, or nominal 0.8V for UltraScale FPGAs. The common mode voltage for UltraScale+ FPGAs is MGTAVCC, or nominal 0.9V. See the UltraScale and UltraScale+ device data sheets (see http://www.xilinx.com/documentation) for exact specifications.

*Figure 53:* **GTM Transceiver Board Design Guidelines**




# GTM Transceiver Reference Clock Checklist

These criteria must be met when choosing an oscillator for a design with GTM transceivers:

- Provide AC coupling between the oscillator output pins and the dedicated GTM transceiver DUAL clock input pins.
- Ensure that the differential voltage swing of the reference clock is the range as specified in the UltraScale+ device data sheets (see http://www.xilinx.com/documentation). The nominal range is 250 mV–2000 mV and the nominal value is 1200 mV).

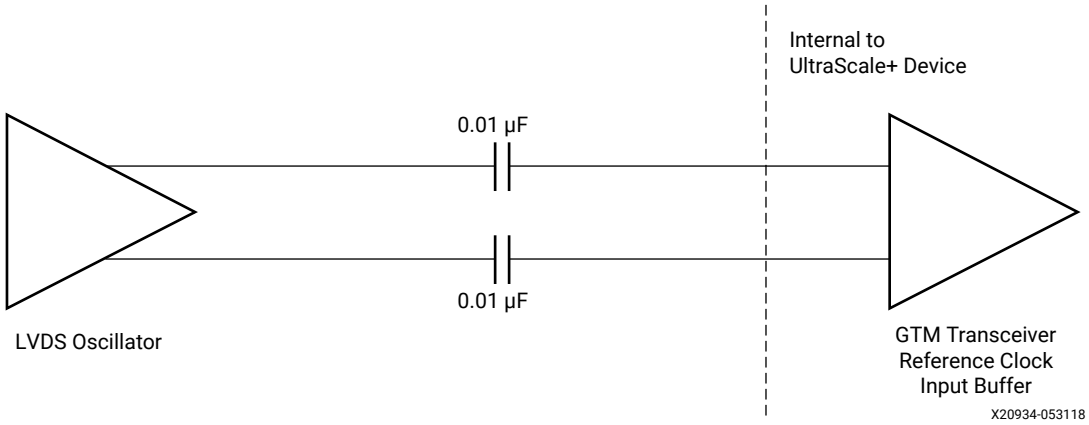
- Meet or exceed the reference clock characteristics as specified in the UltraScale+ device data sheets.
- Meet or exceed the reference clock characteristics as specified in the standard for which the GTM transceiver provides physical layer support.
- Fulfill the oscillator vendor's requirement regarding power supply, board layout, and noise specification.
- Provide a dedicated point-to-point connection between the oscillator and GTM transceiver Dual clock input pins.
- Keep impedance discontinuities on the differential transmission lines to a minimum (impedance discontinuities generate jitter).

# Reference Clock Interface

### LVDS

The following figure shows how an LVDS oscillator is connected to a reference clock input of a GTM transceiver.

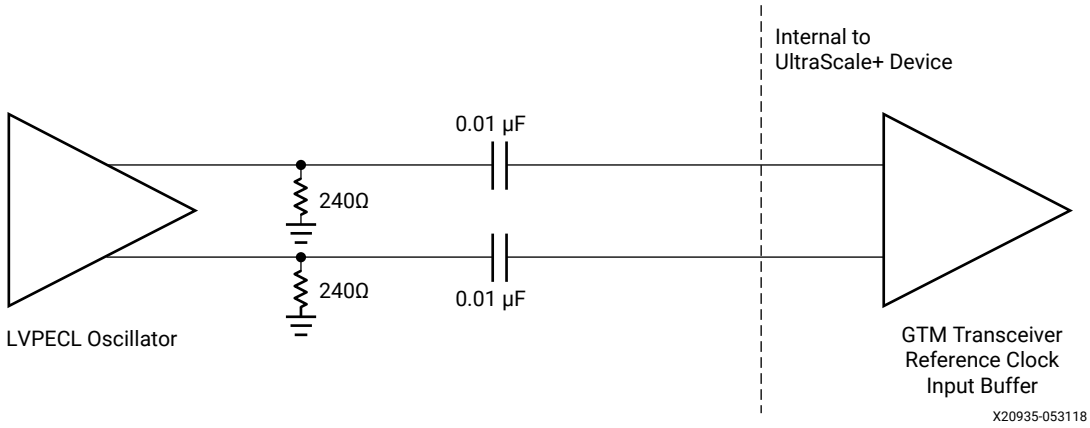
*Figure 54:* **Interfacing an LVDS Oscillator to the GTM Transceiver Reference Clock Input**




### LVPECL

The following figure shows how an LVPECL oscillator is connected to a reference clock input of a GTM transceiver.

*Figure 55:* **Interfacing an LVPECL Oscillator to the GTM Transceiver Reference Clock Input**




Notes relevant to the figure:

1. The resistor values are nominal. Refer to the oscillator data sheet for actual bias resistor requirement.
2. Before completion of device configuration, the termination resistor is not calibrated and the voltage level input to the clock input buffer should be made sure to not exceed the absolute maximum rating as described in the UltraScale+ device data sheets (see http://www.xilinx.com/documentation).

# AC Coupled Reference Clock

AC coupling of the oscillator reference clock output to the GTM transceiver Dual reference clock inputs serves multiple purposes:

- Blocking a DC current between the oscillator and the GTM transceiver Dual dedicated clock input pins (which reduces the power consumption of both parts as well).
- Common mode voltage independence.
- The AC coupling capacitor forms a high-pass filter with the on-chip termination that attenuates wander of the reference clock.

To minimize noise and power consumption, external AC coupling capacitors between the sourcing oscillator and the GTM transceiver Dual dedicated reference clock input pins are required.

# Unused Reference Clocks

If the reference clock input is not used, leave the reference clock input pins unconnected (both MGTREFCLKP and MGTREFCLKN).

# Reference Clock Output Buffer

The reference clock pins can be configured to be output pins that drive an RX recovered clock from one of the transceivers in the Dual. Operation and configuration of this buffer is discussed in Chapter 2: Shared Features. This output is designed to supply a signal through DC blocking capacitors on the PCB. The signal levels are comparable to those of LVDS after the DC blocking capacitors. See the UltraScale+ device data sheets (see http://www.xilinx.com/documentation) for output levels.

# Reference Clock Power

The GTM transceiver reference clock input circuit is powered by MGTAVCC. Excessive noise on this supply has a negative impact on the performance of any GTM transceiver Dual that uses the reference clock from this circuit.

# Power Supply and Filtering

The GTM transceiver Dual requires three analog power supplies: MGTAVCC at a nominal voltage level of 0.85 VDC or 0.9 VDC for UltraScale+ FPGAs, MGTVCCAUX at a nominal voltage level of 1.8 VDC, VCCINT_GT at a nominal voltage of 0.85 VDC, and MGTAVTT at a nominal voltage level of 1.2 VDC. The pins for each of these analog power supplies are tied to a plane in the package. In some packages, there are two planes (a north plane and a south plane) for each of the analog power supplies. See Analog Power Supply Pins for a discussion of the internal power planes in the GTM transceiver packages.

Noise on the GTM transceiver analog power supplies can cause degradation in the performance of the transceivers. The most likely form of degradation is an increase in jitter at the output of the GTM transmitter and reduced jitter tolerance in the receiver. Sources of power supply noise are:

- Power supply regulator noise

- Power distribution network
- Coupling from other circuits

Each of these noise sources must be considered in the design and implementation of the GTM transceiver analog power supplies. The total peak-to-peak noise as measured at the input pin of the UltraScale device should not exceed 10 mVpk-pk.

## Power Up/Down and Reset on Multiple Lanes

The operating state of the GTM transceiver can be controlled through the enabling and disabling of the power down and reset controls (see Reset and Initialization and Power Down). When the GTM transceiver's operating state is changed by either changing the power down state or the reset state, the load current as seen by the on-board power distribution network (PDN) and the power supply regulator is also changed. When the load current changes, the power supply regulator must sense the change in the load current and compensate for this change to maintain the design supply voltage. The effect of a delay in the change in the load current can result in a temporary spike or dip in the power supply voltage. When the operating state of the GTM transceiver goes from power down to power up, the load current transient is positive and the voltage from the regulator might dip while the regulator circuit adapts to the new load conditions. Conversely, when the operating state of the GTM transceiver goes from power up to power down, the load current transient is negative, and the voltage from the regulator might spike while the regulator circuit adapts to the new load current conditions. The magnitude and duration of the voltage transient from the power supply regulator depends upon the design of the power supply regulator circuit. In some cases, the voltage might oscillate as the voltage regulator circuit converges to the design voltage setting. In all of these cases, the important consideration is that the voltage at the input pin of the device must remain within the operating limits as specified in the UltraScale+ device data sheets (see http://www.xilinx.com/documentation). Use the Xilinx Power Estimator (XPE) tool to calculate the amount of power required for the transceivers in your application.

## Power Supply Regulators

Normally, the GTM transceiver analog voltage supplies have local power supply regulators that provide a final stage of voltage regulation. Preferably these regulators are placed as close as is feasible to the GTM transceiver power supply pins. Minimizing the distance between the analog voltage regulators and the GTM transceiver power supply pins reduces the opportunity for noise coupling into the supply after the regulator and for noise generated by current transients caused by load dynamics.

# Linear versus Switching Regulators

The type of power supply regulator can have a significant impact on the complexity, cost, and performance of the power supply circuit. A power supply regulator must provide adequate power to the GTM transceiver with a minimum amount of noise while meeting the overall system thermal and efficiency requirements. There are two major types of power supply voltage regulators available for regulating the GTM transceiver analog voltage rails, linear regulators, and switching regulators. Each of these types of regulator has advantages and disadvantages. The optimal choice of regulator type depends on system requirements such as:

- Physical size
- Thermal budget
- Power efficiency
- Cost

## *Linear Regulator*

A linear regulator is usually the simplest means to provide voltage regulation for the GTM transceiver analog supply rails. Inherently, a linear regulator does not inject significant noise into the regulated output voltage. In fact, some, not all, linear regulators provide noise rejection at the output from noise present on the voltage input. Another advantage of the linear regulator is that it usually requires a minimal number of external components to realize a circuit on the printed circuit board.

There are potentially two major disadvantages to linear regulators: minimum dropout voltage, and limited efficiency. Linear regulators require an input voltage that is higher than the output voltage. This minimum dropout voltage often is dependent on the load current. Even low dropout linear regulators require a minimum difference between the input voltage and the output voltage of the regulator. The system power supply design must consider the minimum dropout voltage requirements of the linear regulators.

The efficiency of a linear regulator is dependent on the voltage difference between the input and output of the linear regulator. For instance, if the input voltage of the regulator is 2.5 VDC and the output voltage of the regulator is 1.2 VDC, the voltage difference is 1.3 VDC. Assuming that the current into the regulator is essentially equal to the current out of the regulator, the maximum efficiency of the regulator is 48%. This means that for every watt delivered to the load, the system must consume an additional watt for regulation. This power consumed by the regulator generates heat that must be dissipated by the system. Providing a means to dissipate the heat generated by the linear regulator can drive up the system cost. So even though from a simple component count and complexity cost the linear regulator appears to have an advantage over the switching regulator, if the overall system cost is considered, including power consumption and heat dissipation, in high current applications the linear regulator can actually be at a disadvantage.

### *Switching Regulator*

A switching regulator can provide an efficient means to deliver a well-regulated voltage for the GTM transceiver analog power supply. Unlike the linear regulator, the switching regulator does not depend on the voltage drop between the input voltage of the regulator and the output voltage to provide regulation. Therefore, the switching regulator can supply large amounts of current to the load while maintaining high power efficiency. It is not uncommon for a switching regulator to maintain efficiencies of 95% or greater. This efficiency is not severely impacted by the voltage drop between the input of the regulator and the output. It is impacted by the load current in a much lesser degree than that of the linear regulator. Because of the efficiency of the switching regulator, the system does not need to supply as much power to the circuit, and it does not need to provide a means to dissipate power consumed by the regulator.

The disadvantages to the switching regulator are complexity of the circuit and noise generated by the regulator switching function. Switching regulator circuits are usually more complex than linear regulator circuits. This shortcoming in switching regulators has recently been addressed by several switching regulator component vendors. Normally, a switching power supply regulation circuit requires a switching transistor element, an inductor, and a capacitor. Depending on the required efficiency and load requirements, a switching regulator circuit might require external switching transistors and inductors. Besides the component count, these switching regulators require very careful placement and routing on the printed circuit board to be effective. Switching regulators generate significant noise and therefore usually require additional filtering before the voltage is delivered to the analog power supply input of the GTM transceiver. Because the amplitude of the noise should be limited to less than 10 mVpp, the power supply filter should be designed to attenuate the noise from the switching regulator to meet this requirement.

## Power Supply Distribution Network

#### Die

The decoupling capacitance on the die filters the highest frequency noise components on the power supplies. The source for this very high frequency noise is the internal on-die circuits.

#### Package

The UltraScale+ architecture package has additional decoupling. Decoupling capacitors in the package provide attenuation for noise in the package power plane, thereby reducing the interaction between GTM transceiver Duals. These capacitors in the package also aid in maintaining a low-impedance, high-frequency path between the power supply, MGTAVCC, MGTVCCAUX, VCCINT_GT or MGTAVTT, and GND.

## Printed Circuit Board

Because the impedance between the power planes and GND has been kept low on the die and in the package, the board design has a much more relaxed requirement for decoupling on the printed circuit board. The primary purpose of the PCB decoupling capacitors is to provide noise isolation between the transceiver power supply pins and the external noise sources. Some examples of external noise sources are:

- Power supply regulator circuits
- On-board digital switching circuits
- SelectIO signals from the UltraScale+ device

Decoupling capacitors should be provided on the PCB near the GTM transceiver power pins. These capacitors reduce the impedance of the PCB power distribution network. The reduced impedance of the PDN provides a means to attenuate noise from external sources before it can get into the device package power planes. The noise at the power pins should be less than 10 mVpp over the band from 10 kHz to 200 MHz.

# PCB Design Checklist

The following table is a checklist of items that can be used to design and review any GTM transceiver PCB schematic and layout.

*Table 74:* **GTM Transceiver PCB Design Checklist**

| Pins | Recommendations |
|---|---|
| MGTREFCLKP<br>MGTREFCLKN | When configured as an input:<br>• Use AC coupling capacitors for connection to oscillator.<br>• For AC coupling capacitors, see Reference Clock Interface.<br>• Reference clock oscillator output must comply with the minimum and maximum input amplitude requirements for these input pins. See the UltraScale+ device data sheets (see http://www.xilinx.com/documentation).<br>• On the printed circuit board, isolation of greater than 65 dB must be maintained between MGTREFCLK nets and SelectIO nets. The isolation should be verified using full 3D electromagnetic (EM) simulations.<br>When configured as an output:<br>• Use AC coupling capacitors for connection to receiving device.<br>• For AC coupling capacitors use 0.01 µF.<br>• For output signal characteristics, see the Virtex UltraScale device data sheet (see http://www.xilinx.com/documentation).<br>• If reference pins are not used, leave the associated pin pair unconnected. However, if the IBUFDS_GTME3/4 is instantiated in the design but not used, the associated pin pair should be connected to GND. |
| MGTMRXP[1:0]<br>MGTMRXN[1:0] | • Use AC coupling capacitors for connection to transmitter. The recommended value for AC coupling capacitors is 100 nF.<br>• Receiver data traces should be provided enough clearance to eliminate crosstalk from adjacent signals.<br>• If a receiver will never be used under any conditions, connect the associated pin pair to GND.<br>• If a receiver is not used and not connected to anything under some conditions, but might be connected to something and used under other conditions, or if a receiver is not used but connected for future use, then for the conditions when the receiver is unused, either do not instance the GTM transceiver in the FPGA design, or if the GTM transceiver is instanced, set RXPD[1:0] to `2'b11`.<br>• See RX Analog Front End. |
| MGTMTXP[1:0]<br>MGTMTXN[1:0] | • Transmitter should be AC coupled to the receiver. The recommended value for the AC coupling capacitors is 100 nF.<br>• Transmitter data traces should be provided enough clearance to eliminate crosstalk from adjacent signals.<br>• If a transmitter is not used, leave the associated pin pair unconnected. |

*Table 74:* **GTM Transceiver PCB Design Checklist** *(cont'd)*

| Pins | Recommendations |
|---|---|
| MGTAVTTRCAL | • Connect to MGTAVTT and to a 100Ω resistor that is also connected to MGTRREF. Use identical trace geometry for the connection between the resistor and this pin, and for the connection from the other pin of the resistor to MGTRREF. Also, the DC resistance of the PCB trace should be limited to less than 0.5Ω.<br>• See Termination Resistor Calibration Circuit.<br>• If an entire PSG is not used by any Duals, tie MGTAVTTRCAL to ground. |
| MGTRREF | • Connect to a 100Ω resistor that is also connected to MGTAVTTRCAL. Use identical trace geometry for the connection between the resistor to this pin, and for the connection from the other pin of the resistor to MGTAVTTRCAL. Also, the DC resistance of the PCB trace should be limited to less than 0.5Ω.<br>• See Termination Resistor Calibration Circuit.<br>• If an entire PSG is not used by any Duals, tie MGTRREF to ground. |
| MGTAVCC[N][1] | • For UltraScale+ FPGAs, the nominal voltage is 0.9 VDC.<br>• See the UltraScale+ device data sheets (see http://www.xilinx.com/documentation) for power supply voltage tolerances.<br>• The power supply regulator for this voltage should not be shared with non-transceiver loads.<br>• Many packages have multiple groups of power supply connections in the package for MGTAVCC. Information on pin locations for each package can be found in the *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification* (UG575).<br>• The following filter capacitor is recommended:<br>◦ 1 of 4.7 µF ± 10% per power supply group (see Figure 49)<br>• For optimal performance, power supply noise must be less than 10 mVpp.<br>• If all of the Duals in a power supply group are not used, the associated power pins can be left unconnected or tied to GND.<br>• For power consumption, refer to the Xilinx Power Estimator (XPE) at www.xilinx.com/power. |

*Table 74:* **GTM Transceiver PCB Design Checklist** *(cont'd)*

| Pins | Recommendations |
|---|---|
| MGTMAVTT[N][1] | • For UltraScale+ FPGAs, the nominal voltage is 1.2 VDC.<br>• See the UltraScale+ device data sheets (see http://www.xilinx.com/documentation) for power supply voltage tolerances.<br>• The power supply regulator for this voltage should not be shared with non-transceiver loads.<br>• Many packages have multiple groups of power supply connections in the package for MGTAVTT. Information on pin locations for each package can be found in the *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification* (UG575).<br>• The following filter capacitor is recommended:<br>◦ 1 of 4.7 µF ± 10% per power supply group (see Figure 49)<br>• For optimal performance, power supply noise must be less than 10 mVpp.<br>• If all of the Duals in a power supply group are not used, the associated power pins can be left unconnected or tied to GND.<br>• For power consumption, refer to the Xilinx Power Estimator (XPE) at www.xilinx.com/power. |
| VCCINT_GT[N][1] | • For UltraScale+ FPGAs, the nominal voltage is 0.85 VDC or 0.9 VDC.<br>• See the UltraScale+ device data sheets (see http://www.xilinx.com/documentation) for power supply voltage tolerances.<br>• Many packages have multiple groups of power supply connections in the package for VCCINT_GT. Information on pin locations for each package can be found in the *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification* (UG575).<br>• For all speed grades other than -2LE, VCCINT and VCCINT_GT can be connected to the same power regulation circuit. Any time VCCINT and VCCINT_GT are connected to separate voltage regulation circuits, VCCINT_GT must be within the operating voltage range as specified in the data sheet prior to device configuration.<br>• The following filter capacitor is recommended:<br>◦ 1 of 4.7 µF ± 10% per power supply group (see Figure 49)<br>• For optimal performance, power supply noise must be less than 10 mVpp.<br>• If all of the Duals in a power supply group are not used, the associated power pins can be left unconnected or tied to GND.<br>• For power consumption, refer to XPE at www.xilinx.com/power. |

*Table 74:* **GTM Transceiver PCB Design Checklist** *(cont'd)*

| Pins | Recommendations |
|---|---|
| MGTVCCAUX[N][1] | • For UltraScale+ FPGAs, the nominal voltage is 1.8 VDC.<br>• See the UltraScale+ device data sheets (see http://www.xilinx.com/documentation) for power supply voltage tolerances.<br>• The power supply regulator for this voltage should not be shared with non-transceiver loads.<br>• Many packages have multiple groups of power supply connections in the package for MGTVCCAUX. Information on pin locations for each package can be found in the *UltraScale and UltraScale+ FPGAs Packaging and Pinouts Product Specification* (UG575).<br>• The following filter capacitor is recommended:<br>◦ 1 of 4.7 µF ± 10% per power supply group (see Figure 49)<br>• For optimal performance, power supply noise must be less than 10 mVpp.<br>• If all of the QPLLs in this power supply group are not used but the Duals are used, the filter capacitors are not necessary. These pins can be connected to $V_{CCAUX}$.<br>• For power consumption, refer to XPE at www.xilinx.com/power. |

**Notes:**

1. N refers to different power supply groups in the package. See Figure 49 for the organization of power supply groups in each package.
2. In -2LE devices, VCCINT = 0.72V, but VCCINT_GT must still be maintained at 0.85V. In all other speed grades, VCCINT and VCCINT_GT are at the same voltage level.

XILINX

*Appendix A*

# DRP Address Map of the GTM Transceiver in UltraScale+ FGPAs

## GTM_DUAL Primitive DRP Address Map

The following table lists the DRP map of the GTM_DUAL primitive sorted by address.

***Note:*** DO NOT modify the reserved bits. Attributes are not described explicitly and are set automatically by the UltraScale FGPAs Transceiver Wizard. These attributes must be left at their defaults, except for use cases that explicitly request different values.

*Table 75:* **DRP Map of GTM_DUAL Primitive**

| DRP Address | DRP Bits | R/W | Attribute Name | Attribute Bits | Attribute Encoding | DRP Encoding |
|---|---|---|---|---|---|---|
| 0x001 | [15:0] | R/W | CH0_TX_DRV_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x002 | [15:0] | R/W | CH0_TX_DRV_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x003 | [15:0] | R/W | CH0_TX_DRV_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x004 | [15:0] | R/W | CH0_TX_ANA_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x005 | [15:0] | R/W | CH0_TX_ANA_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x006 | [15:0] | R/W | CH0_TX_ANA_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x007 | [15:0] | R/W | CH0_TX_DRV_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x008 | [15:0] | R/W | CH0_TX_DRV_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x009 | [15:0] | R/W | CH0_TX_DRV_CFG5 | [15:0] | 0–65535 | 0–65535 |
| 0x00a | [15:0] | R/W | CH0_TX_LPBK_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x00b | [15:0] | R/W | CH0_TX_LPBK_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x00c | [15:0] | R/W | CH0_TX_ANA_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x00d | [15:0] | R/W | CH0_TX_CAL_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x00e | [15:0] | R/W | CH0_TX_CAL_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x00f | [15:0] | R/W | CH0_TX_ANA_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x010 | [15:0] | R/W | CH0_RX_ANA_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x011 | [15:0] | R/W | CH0_RX_ANA_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x012 | [15:0] | R/W | CH0_RX_PAD_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x013 | [15:0] | R/W | CH0_RX_PAD_CFG1 | [15:0] | 0–65535 | 0–65535 |

*Table 75:* **DRP Map of GTM_DUAL Primitive** *(cont'd)*

| DRP Address | DRP Bits | R/W | Attribute Name | Attribute Bits | Attribute Encoding | DRP Encoding |
|---|---|---|---|---|---|---|
| 0x014 | [15:0] | R/W | CH0_RX_CDR_CFG4A | [15:0] | 0–65535 | 0–65535 |
| 0x015 | [15:0] | R/W | CH0_RX_CDR_CFG4B | [15:0] | 0–65535 | 0–65535 |
| 0x016 | [15:0] | R/W | CH0_RX_CDR_CFG3A | [15:0] | 0–65535 | 0–65535 |
| 0x017 | [15:0] | R/W | CH0_RX_CDR_CFG3B | [15:0] | 0–65535 | 0–65535 |
| 0x018 | [15:0] | R/W | CH0_RX_CDR_CFG2A | [15:0] | 0–65535 | 0–65535 |
| 0x019 | [15:0] | R/W | CH0_RX_CDR_CFG2B | [15:0] | 0–65535 | 0–65535 |
| 0x01a | [15:0] | R/W | CH0_RX_CDR_CFG1A | [15:0] | 0–65535 | 0–65535 |
| 0x01b | [15:0] | R/W | CH0_RX_CDR_CFG1B | [15:0] | 0–65535 | 0–65535 |
| 0x01c | [15:0] | R/W | CH0_RX_CDR_CFG0A | [15:0] | 0–65535 | 0–65535 |
| 0x01d | [15:0] | R/W | CH0_RX_CDR_CFG0B | [15:0] | 0–65535 | 0–65535 |
| 0x01e | [15:0] | R/W | CH0_RX_CLKGN_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x01f | [15:0] | R/W | CH0_RX_ANA_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x020 | [15:0] | R/W | CH0_RX_APT_CTRL_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x021 | [15:0] | R/W | CH0_RX_APT_CTRL_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x022 | [15:0] | R/W | CH0_RX_APT_CFG0A | [15:0] | 0–65535 | 0–65535 |
| 0x023 | [15:0] | R/W | CH0_RX_APT_CFG0B | [15:0] | 0–65535 | 0–65535 |
| 0x024 | [15:0] | R/W | CH0_RX_APT_CFG1A | [15:0] | 0–65535 | 0–65535 |
| 0x025 | [15:0] | R/W | CH0_RX_APT_CFG1B | [15:0] | 0–65535 | 0–65535 |
| 0x026 | [15:0] | R/W | CH0_RX_APT_CFG2A | [15:0] | 0–65535 | 0–65535 |
| 0x027 | [15:0] | R/W | CH0_RX_APT_CFG2B | [15:0] | 0–65535 | 0–65535 |
| 0x028 | [15:0] | R/W | CH0_RX_APT_CFG3A | [15:0] | 0–65535 | 0–65535 |
| 0x029 | [15:0] | R/W | CH0_RX_APT_CFG3B | [15:0] | 0–65535 | 0–65535 |
| 0x02a | [15:0] | R/W | CH0_RX_APT_CFG4A | [15:0] | 0–65535 | 0–65535 |
| 0x02b | [15:0] | R/W | CH0_RX_APT_CFG4B | [15:0] | 0–65535 | 0–65535 |
| 0x02c | [15:0] | R/W | CH0_RX_APT_CFG5A | [15:0] | 0–65535 | 0–65535 |
| 0x02d | [15:0] | R/W | CH0_RX_APT_CFG5B | [15:0] | 0–65535 | 0–65535 |
| 0x02e | [15:0] | R/W | CH0_RX_APT_CFG6A | [15:0] | 0–65535 | 0–65535 |
| 0x02f | [15:0] | R/W | CH0_RX_APT_CFG6B | [15:0] | 0–65535 | 0–65535 |
| 0x030 | [15:0] | R/W | CH0_RX_APT_CFG7A | [15:0] | 0–65535 | 0–65535 |
| 0x031 | [15:0] | R/W | CH0_RX_APT_CFG7B | [15:0] | 0–65535 | 0–65535 |
| 0x032 | [15:0] | R/W | CH0_RX_APT_CFG8A | [15:0] | 0–65535 | 0–65535 |
| 0x033 | [15:0] | R/W | CH0_RX_APT_CFG8B | [15:0] | 0–65535 | 0–65535 |
| 0x034 | [15:0] | R/W | CH0_RX_APT_CFG9A | [15:0] | 0–65535 | 0–65535 |
| 0x035 | [15:0] | R/W | CH0_RX_APT_CFG9B | [15:0] | 0–65535 | 0–65535 |
| 0x036 | [15:0] | R/W | CH0_RX_APT_CFG10A | [15:0] | 0–65535 | 0–65535 |
| 0x037 | [15:0] | R/W | CH0_RX_APT_CFG10B | [15:0] | 0–65535 | 0–65535 |
| 0x038 | [15:0] | R/W | CH0_RX_APT_CFG11A | [15:0] | 0–65535 | 0–65535 |
| 0x039 | [15:0] | R/W | CH0_RX_APT_CFG11B | [15:0] | 0–65535 | 0–65535 |

*Table 75:* **DRP Map of GTM_DUAL Primitive** *(cont'd)*

| DRP Address | DRP Bits | R/W | Attribute Name | Attribute Bits | Attribute Encoding | DRP Encoding |
|---|---|---|---|---|---|---|
| 0x03a | [15:0] | R/W | CH0_RX_APT_CFG12A | [15:0] | 0–65535 | 0–65535 |
| 0x03b | [15:0] | R/W | CH0_RX_APT_CFG12B | [15:0] | 0–65535 | 0–65535 |
| 0x03c | [15:0] | R/W | CH0_RX_APT_CFG13A | [15:0] | 0–65535 | 0–65535 |
| 0x03d | [15:0] | R/W | CH0_RX_APT_CFG13B | [15:0] | 0–65535 | 0–65535 |
| 0x03e | [15:0] | R/W | CH0_RX_APT_CFG14A | [15:0] | 0–65535 | 0–65535 |
| 0x03f | [15:0] | R/W | CH0_RX_APT_CFG14B | [15:0] | 0–65535 | 0–65535 |
| 0x040 | [15:0] | R/W | CH0_RX_APT_CFG15A | [15:0] | 0–65535 | 0–65535 |
| 0x041 | [15:0] | R/W | CH0_RX_APT_CFG15B | [15:0] | 0–65535 | 0–65535 |
| 0x042 | [15:0] | R/W | CH0_RX_APT_CFG16A | [15:0] | 0–65535 | 0–65535 |
| 0x043 | [15:0] | R/W | CH0_RX_APT_CFG16B | [15:0] | 0–65535 | 0–65535 |
| 0x044 | [15:0] | R/W | CH0_RX_APT_CFG17A | [15:0] | 0–65535 | 0–65535 |
| 0x045 | [15:0] | R/W | CH0_RX_APT_CFG17B | [15:0] | 0–65535 | 0–65535 |
| 0x046 | [15:0] | R/W | CH0_RX_APT_CFG18A | [15:0] | 0–65535 | 0–65535 |
| 0x047 | [15:0] | R/W | CH0_RX_APT_CFG18B | [15:0] | 0–65535 | 0–65535 |
| 0x048 | [15:0] | R/W | CH0_RX_APT_CFG19A | [15:0] | 0–65535 | 0–65535 |
| 0x049 | [15:0] | R/W | CH0_RX_APT_CFG19B | [15:0] | 0–65535 | 0–65535 |
| 0x04a | [15:0] | R/W | CH0_RX_APT_CFG20A | [15:0] | 0–65535 | 0–65535 |
| 0x04b | [15:0] | R/W | CH0_RX_APT_CFG20B | [15:0] | 0–65535 | 0–65535 |
| 0x04c | [15:0] | R/W | CH0_RX_APT_CFG21A | [15:0] | 0–65535 | 0–65535 |
| 0x04d | [15:0] | R/W | CH0_RX_APT_CFG21B | [15:0] | 0–65535 | 0–65535 |
| 0x04e | [15:0] | R/W | CH0_RX_APT_CFG28A | [15:0] | 0–65535 | 0–65535 |
| 0x04f | [15:0] | R/W | CH0_RX_APT_CFG28B | [15:0] | 0–65535 | 0–65535 |
| 0x050 | [15:0] | R/W | CH0_RX_APT_CFG22A | [15:0] | 0–65535 | 0–65535 |
| 0x051 | [15:0] | R/W | CH0_RX_APT_CFG22B | [15:0] | 0–65535 | 0–65535 |
| 0x052 | [15:0] | R/W | CH0_RX_APT_CFG23A | [15:0] | 0–65535 | 0–65535 |
| 0x053 | [15:0] | R/W | CH0_RX_APT_CFG23B | [15:0] | 0–65535 | 0–65535 |
| 0x054 | [15:0] | R/W | CH0_RX_APT_CFG24A | [15:0] | 0–65535 | 0–65535 |
| 0x055 | [15:0] | R/W | CH0_RX_APT_CFG24B | [15:0] | 0–65535 | 0–65535 |
| 0x056 | [15:0] | R/W | CH0_RX_APT_CFG25A | [15:0] | 0–65535 | 0–65535 |
| 0x057 | [15:0] | R/W | CH0_RX_APT_CFG25B | [15:0] | 0–65535 | 0–65535 |
| 0x058 | [15:0] | R/W | CH0_RX_APT_CFG26A | [15:0] | 0–65535 | 0–65535 |
| 0x059 | [15:0] | R/W | CH0_RX_APT_CFG26B | [15:0] | 0–65535 | 0–65535 |
| 0x05a | [15:0] | R/W | CH0_RX_APT_CFG27A | [15:0] | 0–65535 | 0–65535 |
| 0x05b | [15:0] | R/W | CH0_RX_APT_CFG27B | [15:0] | 0–65535 | 0–65535 |
| 0x05c | [15:0] | R/W | CH0_RX_DSP_CFG | [15:0] | 0–65535 | 0–65535 |
| 0x064 | [15:0] | R/W | CH0_RX_CAL_CFG2A | [15:0] | 0–65535 | 0–65535 |
| 0x065 | [15:0] | R/W | CH0_RX_CAL_CFG2B | [15:0] | 0–65535 | 0–65535 |
| 0x067 | [15:0] | R/W | CH0_RX_CAL_CFG0A | [15:0] | 0–65535 | 0–65535 |

*Table 75:* **DRP Map of GTM_DUAL Primitive** *(cont'd)*

| DRP Address | DRP Bits | R/W | Attribute Name | Attribute Bits | Attribute Encoding | DRP Encoding |
|---|---|---|---|---|---|---|
| 0x068 | [15:0] | R/W | CH0_RX_CAL_CFG0B | [15:0] | 0–65535 | 0–65535 |
| 0x069 | [15:0] | R/W | CH0_RX_CAL_CFG1A | [15:0] | 0–65535 | 0–65535 |
| 0x06a | [15:0] | R/W | CH0_RX_CAL_CFG1B | [15:0] | 0–65535 | 0–65535 |
| 0x06b | [15:0] | R/W | CH0_RX_ADC_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x06c | [15:0] | R/W | CH0_RX_ADC_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x06e | [15:0] | R/W | CH0_RX_CLKGN_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x06f | [15:0] | R/W | CH0_RX_CTLE_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x070 | [15:0] | R/W | CH0_RX_CTLE_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x071 | [15:0] | R/W | CH0_RX_CTLE_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x072 | [15:0] | R/W | CH0_RX_CTLE_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x080 | [15:0] | R/W | CH0_RX_PCS_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x081 | [15:0] | R/W | CH0_RX_PCS_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x082 | [15:0] | R/W | CH0_RX_MON_CFG | [15:0] | 0–65535 | 0–65535 |
| 0x083 | [15:0] | R/W | CH0_TX_PCS_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x084 | [15:0] | R/W | CH0_TX_PCS_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x085 | [15:0] | R/W | CH0_TX_PCS_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x086 | [15:0] | R/W | CH0_TX_PCS_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x087 | [15:0] | R/W | CH0_TX_PCS_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x088 | [15:0] | R/W | CH0_TX_PCS_CFG5 | [15:0] | 0–65535 | 0–65535 |
| 0x089 | [15:0] | R/W | CH0_TX_PCS_CFG6 | [15:0] | 0–65535 | 0–65535 |
| 0x08a | [15:0] | R/W | CH0_TX_PCS_CFG7 | [15:0] | 0–65535 | 0–65535 |
| 0x08b | [15:0] | R/W | CH0_TX_PCS_CFG8 | [15:0] | 0–65535 | 0–65535 |
| 0x08c | [15:0] | R/W | CH0_TX_PCS_CFG9 | [15:0] | 0–65535 | 0–65535 |
| 0x08d | [15:0] | R/W | CH0_TX_PCS_CFG10 | [15:0] | 0–65535 | 0–65535 |
| 0x08e | [15:0] | R/W | CH0_TX_PCS_CFG11 | [15:0] | 0–65535 | 0–65535 |
| 0x08f | [15:0] | R/W | CH0_TX_PCS_CFG12 | [15:0] | 0–65535 | 0–65535 |
| 0x090 | [15:0] | R/W | CH0_TX_PCS_CFG13 | [15:0] | 0–65535 | 0–65535 |
| 0x091 | [15:0] | R/W | CH0_TX_PCS_CFG14 | [15:0] | 0–65535 | 0–65535 |
| 0x092 | [15:0] | R/W | CH0_TX_PCS_CFG15 | [15:0] | 0–65535 | 0–65535 |
| 0x093 | [15:0] | R/W | CH0_TX_PCS_CFG16 | [15:0] | 0–65535 | 0–65535 |
| 0x094 | [15:0] | R/W | CH0_TX_PCS_CFG17 | [15:0] | 0–65535 | 0–65535 |
| 0x095 | [15:0] | R/W | CH0_A_CH_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x096 | [15:0] | R/W | CH0_A_CH_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x097 | [15:0] | R/W | CH0_A_CH_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x098 | [15:0] | R/W | CH0_A_CH_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x099 | [15:0] | R/W | CH0_A_CH_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x09a | [15:0] | R/W | CH0_A_CH_CFG5 | [15:0] | 0–65535 | 0–65535 |
| 0x09b | [15:0] | R/W | CH0_A_CH_CFG6 | [15:0] | 0–65535 | 0–65535 |

*Table 75:* **DRP Map of GTM_DUAL Primitive** *(cont'd)*

| DRP Address | DRP Bits | R/W | Attribute Name | Attribute Bits | Attribute Encoding | DRP Encoding |
|---|---|---|---|---|---|---|
| 0x0a0 | [15:0] | R/W | CH0_RST_TIME_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x0a1 | [15:0] | R/W | CH0_RST_TIME_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x0a2 | [15:0] | R/W | CH0_RST_TIME_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x0a3 | [15:0] | R/W | CH0_RST_TIME_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x0a4 | [15:0] | R/W | CH0_RST_TIME_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x0a5 | [15:0] | R/W | CH0_RST_TIME_CFG5 | [15:0] | 0–65535 | 0–65535 |
| 0x0a6 | [15:0] | R/W | CH0_RST_TIME_CFG6 | [15:0] | 0–65535 | 0–65535 |
| 0x0a7 | [15:0] | R/W | CH0_RST_LP_ID_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x0a8 | [15:0] | R/W | CH0_RST_LP_ID_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x0a9 | [15:0] | R/W | CH0_RST_LP_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x0aa | [15:0] | R/W | CH0_RST_LP_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x0ab | [15:0] | R/W | CH0_RST_LP_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x0ac | [15:0] | R/W | CH0_RST_LP_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x0ad | [15:0] | R/W | CH0_RST_LP_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x0ae | [15:0] | R/W | RST_CFG | [15:0] | 0–65535 | 0–65535 |
| 0x0af | [15:0] | R/W | RST_PLL_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x0b0 | [15:0] | R/W | PLL_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x0b1 | [15:0] | R/W | PLL_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x0b2 | [15:0] | R/W | PLL_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x0b3 | [15:0] | R/W | PLL_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x0b4 | [15:0] | R/W | PLL_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x0b5 | [15:0] | R/W | PLL_CFG5 | [15:0] | 0–65535 | 0–65535 |
| 0x0b6 | [15:0] | R/W | PLL_CFG6 | [15:0] | 0–65535 | 0–65535 |
| 0x0b7 | [15:0] | R/W | SDM_SEED_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x0b8 | [15:0] | R/W | SDM_SEED_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x0b9 | [15:0] | R/W | SDM_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x0ba | [15:0] | R/W | SDM_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x0bb | [15:0] | R/W | SDM_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x0bc | [15:0] | R/W | A_SDM_DATA_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x0bd | [15:0] | R/W | A_SDM_DATA_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x0be | [15:0] | R/W | PLL_CRS_CTRL_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x0bf | [15:0] | R/W | PLL_CRS_CTRL_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x0c0 | [15:0] | R/W | A_CFG | [15:0] | 0–65535 | 0–65535 |
| 0x0c1 | [15:0] | R/W | SAP_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x0c3 | [15:0] | R/W | DRPEN_CFG | [15:0] | 0–65535 | 0–65535 |
| 0x0d0 | [15:0] | R/W | FEC_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x0d1 | [15:0] | R/W | FEC_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x0d2 | [15:0] | R/W | FEC_CFG2 | [15:0] | 0–65535 | 0–65535 |

*Table 75:* **DRP Map of GTM_DUAL Primitive** *(cont'd)*

| DRP Address | DRP Bits | R/W | Attribute Name | Attribute Bits | Attribute Encoding | DRP Encoding |
|---|---|---|---|---|---|---|
| 0x0d3 | [15:0] | R/W | FEC_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x0d4 | [15:0] | R/W | FEC_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x0d5 | [15:0] | R/W | FEC_CFG5 | [15:0] | 0–65535 | 0–65535 |
| 0x0d6 | [15:0] | R/W | FEC_CFG6 | [15:0] | 0–65535 | 0–65535 |
| 0x0d7 | [15:0] | R/W | FEC_CFG7 | [15:0] | 0–65535 | 0–65535 |
| 0x0d8 | [15:0] | R/W | FEC_CFG8 | [15:0] | 0–65535 | 0–65535 |
| 0x0d9 | [15:0] | R/W | FEC_CFG9 | [15:0] | 0–65535 | 0–65535 |
| 0x0da | [15:0] | R/W | FEC_CFG10 | [15:0] | 0–65535 | 0–65535 |
| 0x0db | [15:0] | R/W | FEC_CFG11 | [15:0] | 0–65535 | 0–65535 |
| 0x0dc | [15:0] | R/W | FEC_CFG12 | [15:0] | 0–65535 | 0–65535 |
| 0x0dd | [15:0] | R/W | FEC_CFG13 | [15:0] | 0–65535 | 0–65535 |
| 0x0de | [15:0] | R/W | FEC_CFG14 | [15:0] | 0–65535 | 0–65535 |
| 0x0df | [15:0] | R/W | FEC_CFG15 | [15:0] | 0–65535 | 0–65535 |
| 0x0e0 | [15:0] | R/W | FEC_CFG16 | [15:0] | 0–65535 | 0–65535 |
| 0x0e1 | [15:0] | R/W | FEC_CFG17 | [15:0] | 0–65535 | 0–65535 |
| 0x0e2 | [15:0] | R/W | FEC_CFG18 | [15:0] | 0–65535 | 0–65535 |
| 0x0e3 | [15:0] | R/W | FEC_CFG19 | [15:0] | 0–65535 | 0–65535 |
| 0x0e4 | [15:0] | R/W | FEC_CFG20 | [15:0] | 0–65535 | 0–65535 |
| 0x0e5 | [15:0] | R/W | FEC_CFG21 | [15:0] | 0–65535 | 0–65535 |
| 0x0e6 | [15:0] | R/W | FEC_CFG22 | [15:0] | 0–65535 | 0–65535 |
| 0x0e7 | [15:0] | R/W | FEC_CFG23 | [15:0] | 0–65535 | 0–65535 |
| 0x0e8 | [15:0] | R/W | FEC_CFG24 | [15:0] | 0–65535 | 0–65535 |
| 0x0e9 | [15:0] | R/W | FEC_CFG25 | [15:0] | 0–65535 | 0–65535 |
| 0x0ea | [15:0] | R/W | FEC_CFG26 | [15:0] | 0–65535 | 0–65535 |
| 0x0eb | [15:0] | R/W | FEC_CFG27 | [15:0] | 0–65535 | 0–65535 |
| 0x15e | [15:0] | R/W | BIAS_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x15f | [15:0] | R/W | BIAS_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x160 | [15:0] | R/W | BIAS_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x161 | [15:0] | R/W | BIAS_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x162 | [15:0] | R/W | BIAS_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x163 | [15:0] | R/W | BIAS_CFG5 | [15:0] | 0–65535 | 0–65535 |
| 0x164 | [15:0] | R/W | BIAS_CFG6 | [15:0] | 0–65535 | 0–65535 |
| 0x165 | [15:0] | R/W | BIAS_CFG7 | [15:0] | 0–65535 | 0–65535 |
| 0x201 | [15:0] | R/W | CH1_TX_DRV_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x202 | [15:0] | R/W | CH1_TX_DRV_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x203 | [15:0] | R/W | CH1_TX_DRV_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x204 | [15:0] | R/W | CH1_TX_ANA_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x205 | [15:0] | R/W | CH1_TX_ANA_CFG2 | [15:0] | 0–65535 | 0–65535 |

*Table 75:* **DRP Map of GTM_DUAL Primitive** *(cont'd)*

| DRP Address | DRP Bits | R/W | Attribute Name | Attribute Bits | Attribute Encoding | DRP Encoding |
|---|---|---|---|---|---|---|
| 0x206 | [15:0] | R/W | CH1_TX_ANA_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x207 | [15:0] | R/W | CH1_TX_DRV_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x208 | [15:0] | R/W | CH1_TX_DRV_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x209 | [15:0] | R/W | CH1_TX_DRV_CFG5 | [15:0] | 0–65535 | 0–65535 |
| 0x20a | [15:0] | R/W | CH1_TX_LPBK_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x20b | [15:0] | R/W | CH1_TX_LPBK_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x20c | [15:0] | R/W | CH1_TX_ANA_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x20d | [15:0] | R/W | CH1_TX_CAL_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x20e | [15:0] | R/W | CH1_TX_CAL_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x20f | [15:0] | R/W | CH1_TX_ANA_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x210 | [15:0] | R/W | CH1_RX_ANA_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x211 | [15:0] | R/W | CH1_RX_ANA_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x212 | [15:0] | R/W | CH1_RX_PAD_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x213 | [15:0] | R/W | CH1_RX_PAD_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x214 | [15:0] | R/W | CH1_RX_CDR_CFG4A | [15:0] | 0–65535 | 0–65535 |
| 0x215 | [15:0] | R/W | CH1_RX_CDR_CFG4B | [15:0] | 0–65535 | 0–65535 |
| 0x216 | [15:0] | R/W | CH1_RX_CDR_CFG3A | [15:0] | 0–65535 | 0–65535 |
| 0x217 | [15:0] | R/W | CH1_RX_CDR_CFG3B | [15:0] | 0–65535 | 0–65535 |
| 0x218 | [15:0] | R/W | CH1_RX_CDR_CFG2A | [15:0] | 0–65535 | 0–65535 |
| 0x219 | [15:0] | R/W | CH1_RX_CDR_CFG2B | [15:0] | 0–65535 | 0–65535 |
| 0x21a | [15:0] | R/W | CH1_RX_CDR_CFG1A | [15:0] | 0–65535 | 0–65535 |
| 0x21b | [15:0] | R/W | CH1_RX_CDR_CFG1B | [15:0] | 0–65535 | 0–65535 |
| 0x21c | [15:0] | R/W | CH1_RX_CDR_CFG0A | [15:0] | 0–65535 | 0–65535 |
| 0x21d | [15:0] | R/W | CH1_RX_CDR_CFG0B | [15:0] | 0–65535 | 0–65535 |
| 0x21e | [15:0] | R/W | CH1_RX_CLKGN_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x21f | [15:0] | R/W | CH1_RX_ANA_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x220 | [15:0] | R/W | CH1_RX_APT_CTRL_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x221 | [15:0] | R/W | CH1_RX_APT_CTRL_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x222 | [15:0] | R/W | CH1_RX_APT_CFG0A | [15:0] | 0–65535 | 0–65535 |
| 0x223 | [15:0] | R/W | CH1_RX_APT_CFG0B | [15:0] | 0–65535 | 0–65535 |
| 0x224 | [15:0] | R/W | CH1_RX_APT_CFG1A | [15:0] | 0–65535 | 0–65535 |
| 0x225 | [15:0] | R/W | CH1_RX_APT_CFG1B | [15:0] | 0–65535 | 0–65535 |
| 0x226 | [15:0] | R/W | CH1_RX_APT_CFG2A | [15:0] | 0–65535 | 0–65535 |
| 0x227 | [15:0] | R/W | CH1_RX_APT_CFG2B | [15:0] | 0–65535 | 0–65535 |
| 0x228 | [15:0] | R/W | CH1_RX_APT_CFG3A | [15:0] | 0–65535 | 0–65535 |
| 0x229 | [15:0] | R/W | CH1_RX_APT_CFG3B | [15:0] | 0–65535 | 0–65535 |
| 0x22a | [15:0] | R/W | CH1_RX_APT_CFG4A | [15:0] | 0–65535 | 0–65535 |
| 0x22b | [15:0] | R/W | CH1_RX_APT_CFG4B | [15:0] | 0–65535 | 0–65535 |

*Table 75:* **DRP Map of GTM_DUAL Primitive** *(cont'd)*

| DRP Address | DRP Bits | R/W | Attribute Name | Attribute Bits | Attribute Encoding | DRP Encoding |
|---|---|---|---|---|---|---|
| 0x22c | [15:0] | R/W | CH1_RX_APT_CFG5A | [15:0] | 0–65535 | 0–65535 |
| 0x22d | [15:0] | R/W | CH1_RX_APT_CFG5B | [15:0] | 0–65535 | 0–65535 |
| 0x22e | [15:0] | R/W | CH1_RX_APT_CFG6A | [15:0] | 0–65535 | 0–65535 |
| 0x22f | [15:0] | R/W | CH1_RX_APT_CFG6B | [15:0] | 0–65535 | 0–65535 |
| 0x230 | [15:0] | R/W | CH1_RX_APT_CFG7A | [15:0] | 0–65535 | 0–65535 |
| 0x231 | [15:0] | R/W | CH1_RX_APT_CFG7B | [15:0] | 0–65535 | 0–65535 |
| 0x232 | [15:0] | R/W | CH1_RX_APT_CFG8A | [15:0] | 0–65535 | 0–65535 |
| 0x233 | [15:0] | R/W | CH1_RX_APT_CFG8B | [15:0] | 0–65535 | 0–65535 |
| 0x234 | [15:0] | R/W | CH1_RX_APT_CFG9A | [15:0] | 0–65535 | 0–65535 |
| 0x235 | [15:0] | R/W | CH1_RX_APT_CFG9B | [15:0] | 0–65535 | 0–65535 |
| 0x236 | [15:0] | R/W | CH1_RX_APT_CFG10A | [15:0] | 0–65535 | 0–65535 |
| 0x237 | [15:0] | R/W | CH1_RX_APT_CFG10B | [15:0] | 0–65535 | 0–65535 |
| 0x238 | [15:0] | R/W | CH1_RX_APT_CFG11A | [15:0] | 0–65535 | 0–65535 |
| 0x239 | [15:0] | R/W | CH1_RX_APT_CFG11B | [15:0] | 0–65535 | 0–65535 |
| 0x23a | [15:0] | R/W | CH1_RX_APT_CFG12A | [15:0] | 0–65535 | 0–65535 |
| 0x23b | [15:0] | R/W | CH1_RX_APT_CFG12B | [15:0] | 0–65535 | 0–65535 |
| 0x23c | [15:0] | R/W | CH1_RX_APT_CFG13A | [15:0] | 0–65535 | 0–65535 |
| 0x23d | [15:0] | R/W | CH1_RX_APT_CFG13B | [15:0] | 0–65535 | 0–65535 |
| 0x23e | [15:0] | R/W | CH1_RX_APT_CFG14A | [15:0] | 0–65535 | 0–65535 |
| 0x23f | [15:0] | R/W | CH1_RX_APT_CFG14B | [15:0] | 0–65535 | 0–65535 |
| 0x240 | [15:0] | R/W | CH1_RX_APT_CFG15A | [15:0] | 0–65535 | 0–65535 |
| 0x241 | [15:0] | R/W | CH1_RX_APT_CFG15B | [15:0] | 0–65535 | 0–65535 |
| 0x242 | [15:0] | R/W | CH1_RX_APT_CFG16A | [15:0] | 0–65535 | 0–65535 |
| 0x243 | [15:0] | R/W | CH1_RX_APT_CFG16B | [15:0] | 0–65535 | 0–65535 |
| 0x244 | [15:0] | R/W | CH1_RX_APT_CFG17A | [15:0] | 0–65535 | 0–65535 |
| 0x245 | [15:0] | R/W | CH1_RX_APT_CFG17B | [15:0] | 0–65535 | 0–65535 |
| 0x246 | [15:0] | R/W | CH1_RX_APT_CFG18A | [15:0] | 0–65535 | 0–65535 |
| 0x247 | [15:0] | R/W | CH1_RX_APT_CFG18B | [15:0] | 0–65535 | 0–65535 |
| 0x248 | [15:0] | R/W | CH1_RX_APT_CFG19A | [15:0] | 0–65535 | 0–65535 |
| 0x249 | [15:0] | R/W | CH1_RX_APT_CFG19B | [15:0] | 0–65535 | 0–65535 |
| 0x24a | [15:0] | R/W | CH1_RX_APT_CFG20A | [15:0] | 0–65535 | 0–65535 |
| 0x24b | [15:0] | R/W | CH1_RX_APT_CFG20B | [15:0] | 0–65535 | 0–65535 |
| 0x24c | [15:0] | R/W | CH1_RX_APT_CFG21A | [15:0] | 0–65535 | 0–65535 |
| 0x24d | [15:0] | R/W | CH1_RX_APT_CFG21B | [15:0] | 0–65535 | 0–65535 |
| 0x24e | [15:0] | R/W | CH1_RX_APT_CFG28A | [15:0] | 0–65535 | 0–65535 |
| 0x24f | [15:0] | R/W | CH1_RX_APT_CFG28B | [15:0] | 0–65535 | 0–65535 |
| 0x250 | [15:0] | R/W | CH1_RX_APT_CFG22A | [15:0] | 0–65535 | 0–65535 |
| 0x251 | [15:0] | R/W | CH1_RX_APT_CFG22B | [15:0] | 0–65535 | 0–65535 |

*Table 75:* **DRP Map of GTM_DUAL Primitive** *(cont'd)*

| DRP Address | DRP Bits | R/W | Attribute Name | Attribute Bits | Attribute Encoding | DRP Encoding |
|---|---|---|---|---|---|---|
| 0x252 | [15:0] | R/W | CH1_RX_APT_CFG23A | [15:0] | 0–65535 | 0–65535 |
| 0x253 | [15:0] | R/W | CH1_RX_APT_CFG23B | [15:0] | 0–65535 | 0–65535 |
| 0x254 | [15:0] | R/W | CH1_RX_APT_CFG24A | [15:0] | 0–65535 | 0–65535 |
| 0x255 | [15:0] | R/W | CH1_RX_APT_CFG24B | [15:0] | 0–65535 | 0–65535 |
| 0x256 | [15:0] | R/W | CH1_RX_APT_CFG25A | [15:0] | 0–65535 | 0–65535 |
| 0x257 | [15:0] | R/W | CH1_RX_APT_CFG25B | [15:0] | 0–65535 | 0–65535 |
| 0x258 | [15:0] | R/W | CH1_RX_APT_CFG26A | [15:0] | 0–65535 | 0–65535 |
| 0x259 | [15:0] | R/W | CH1_RX_APT_CFG26B | [15:0] | 0–65535 | 0–65535 |
| 0x25a | [15:0] | R/W | CH1_RX_APT_CFG27A | [15:0] | 0–65535 | 0–65535 |
| 0x25b | [15:0] | R/W | CH1_RX_APT_CFG27B | [15:0] | 0–65535 | 0–65535 |
| 0x25c | [15:0] | R/W | CH1_RX_DSP_CFG | [15:0] | 0–65535 | 0–65535 |
| 0x264 | [15:0] | R/W | CH1_RX_CAL_CFG2A | [15:0] | 0–65535 | 0–65535 |
| 0x265 | [15:0] | R/W | CH1_RX_CAL_CFG2B | [15:0] | 0–65535 | 0–65535 |
| 0x267 | [15:0] | R/W | CH1_RX_CAL_CFG0A | [15:0] | 0–65535 | 0–65535 |
| 0x268 | [15:0] | R/W | CH1_RX_CAL_CFG0B | [15:0] | 0–65535 | 0–65535 |
| 0x269 | [15:0] | R/W | CH1_RX_CAL_CFG1A | [15:0] | 0–65535 | 0–65535 |
| 0x26a | [15:0] | R/W | CH1_RX_CAL_CFG1B | [15:0] | 0–65535 | 0–65535 |
| 0x26b | [15:0] | R/W | CH1_RX_ADC_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x26c | [15:0] | R/W | CH1_RX_ADC_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x26e | [15:0] | R/W | CH1_RX_CLKGN_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x26f | [15:0] | R/W | CH1_RX_CTLE_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x270 | [15:0] | R/W | CH1_RX_CTLE_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x271 | [15:0] | R/W | CH1_RX_CTLE_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x272 | [15:0] | R/W | CH1_RX_CTLE_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x280 | [15:0] | R/W | CH1_RX_PCS_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x281 | [15:0] | R/W | CH1_RX_PCS_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x282 | [15:0] | R/W | CH1_RX_MON_CFG | [15:0] | 0–65535 | 0–65535 |
| 0x283 | [15:0] | R/W | CH1_TX_PCS_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x284 | [15:0] | R/W | CH1_TX_PCS_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x285 | [15:0] | R/W | CH1_TX_PCS_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x286 | [15:0] | R/W | CH1_TX_PCS_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x287 | [15:0] | R/W | CH1_TX_PCS_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x288 | [15:0] | R/W | CH1_TX_PCS_CFG5 | [15:0] | 0–65535 | 0–65535 |
| 0x289 | [15:0] | R/W | CH1_TX_PCS_CFG6 | [15:0] | 0–65535 | 0–65535 |
| 0x28a | [15:0] | R/W | CH1_TX_PCS_CFG7 | [15:0] | 0–65535 | 0–65535 |
| 0x28b | [15:0] | R/W | CH1_TX_PCS_CFG8 | [15:0] | 0–65535 | 0–65535 |
| 0x28c | [15:0] | R/W | CH1_TX_PCS_CFG9 | [15:0] | 0–65535 | 0–65535 |
| 0x28d | [15:0] | R/W | CH1_TX_PCS_CFG10 | [15:0] | 0–65535 | 0–65535 |

*Table 75:* **DRP Map of GTM_DUAL Primitive** *(cont'd)*

| DRP Address | DRP Bits | R/W | Attribute Name | Attribute Bits | Attribute Encoding | DRP Encoding |
|---|---|---|---|---|---|---|
| 0x28e | [15:0] | R/W | CH1_TX_PCS_CFG11 | [15:0] | 0–65535 | 0–65535 |
| 0x28f | [15:0] | R/W | CH1_TX_PCS_CFG12 | [15:0] | 0–65535 | 0–65535 |
| 0x290 | [15:0] | R/W | CH1_TX_PCS_CFG13 | [15:0] | 0–65535 | 0–65535 |
| 0x291 | [15:0] | R/W | CH1_TX_PCS_CFG14 | [15:0] | 0–65535 | 0–65535 |
| 0x292 | [15:0] | R/W | CH1_TX_PCS_CFG15 | [15:0] | 0–65535 | 0–65535 |
| 0x293 | [15:0] | R/W | CH1_TX_PCS_CFG16 | [15:0] | 0–65535 | 0–65535 |
| 0x294 | [15:0] | R/W | CH1_TX_PCS_CFG17 | [15:0] | 0–65535 | 0–65535 |
| 0x295 | [15:0] | R/W | CH1_A_CH_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x296 | [15:0] | R/W | CH1_A_CH_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x297 | [15:0] | R/W | CH1_A_CH_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x298 | [15:0] | R/W | CH1_A_CH_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x299 | [15:0] | R/W | CH1_A_CH_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x29a | [15:0] | R/W | CH1_A_CH_CFG5 | [15:0] | 0–65535 | 0–65535 |
| 0x29b | [15:0] | R/W | CH1_A_CH_CFG6 | [15:0] | 0–65535 | 0–65535 |
| 0x2a0 | [15:0] | R/W | CH1_RST_TIME_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x2a1 | [15:0] | R/W | CH1_RST_TIME_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x2a2 | [15:0] | R/W | CH1_RST_TIME_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x2a3 | [15:0] | R/W | CH1_RST_TIME_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x2a4 | [15:0] | R/W | CH1_RST_TIME_CFG4 | [15:0] | 0–65535 | 0–65535 |
| 0x2a5 | [15:0] | R/W | CH1_RST_TIME_CFG5 | [15:0] | 0–65535 | 0–65535 |
| 0x2a6 | [15:0] | R/W | CH1_RST_TIME_CFG6 | [15:0] | 0–65535 | 0–65535 |
| 0x2a7 | [15:0] | R/W | CH1_RST_LP_ID_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x2a8 | [15:0] | R/W | CH1_RST_LP_ID_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x2a9 | [15:0] | R/W | CH1_RST_LP_CFG0 | [15:0] | 0–65535 | 0–65535 |
| 0x2aa | [15:0] | R/W | CH1_RST_LP_CFG1 | [15:0] | 0–65535 | 0–65535 |
| 0x2ab | [15:0] | R/W | CH1_RST_LP_CFG2 | [15:0] | 0–65535 | 0–65535 |
| 0x2ac | [15:0] | R/W | CH1_RST_LP_CFG3 | [15:0] | 0–65535 | 0–65535 |
| 0x2ad | [15:0] | R/W | CH1_RST_LP_CFG4 | [15:0] | 0–65535 | 0–65535 |

XILINX

*Appendix B*

# Additional Resources and Legal Notices

## Xilinx Resources

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

## Documentation Navigator and Design Hubs

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

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

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

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

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

# Please Read: Important Legal Notices

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

**AUTOMOTIVE APPLICATIONS DISCLAIMER**

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

This document contains preliminary information and is subject to change without notice. Information provided herein relates to products and/or services not yet available for sale, and provided solely for information purposes and are not intended, or to be construed, as an offer for sale or an attempted commercialization of the products and/or services referred to herein.

## Copyright