Intel Agilex® 5 FPGA GTS Transceiver Architecture and PMA and FEC Direct PHY IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **23.4**

IP Version: **2.0.0**

Send Feedback

772104
2023.12.18

Contents

1. GTS Transceiver Overview

This user guide describes the architecture and implementation details about the GTS transceivers in Intel Agilex® 5 FPGAs.

- Architecture details of the GTS transceiver in chapter 2
- Implementation details of the GTS IPs in chapter 3 to 6.

The GTS transceivers have a non-return-to-zero (NRZ) serial interface with an advanced physical medium attachment (PMA) and multiple hard IPs to allow efficient implementation of popular and emerging serial protocols. The GTS transceiver banks are monolithically integrated to the FPGA core for greater efficiency in lower power consumption and smaller form factor.

Table 1. Key GTS Transceiver Features

Feature	E-Series (Device Group B)	E-Series (Device Group A)	D-Series
Number of available PMAs	4-24 PMAs		8-32 PMAs
Data rate range	1-17.16 Gbps NRZ	1-28.1 Gbps NRZ	
PCIe* hard IP	Up to six PCIe 3.0 x4 or PCIe 4.0 x4(1)	Up to six PCIe 4.0 x4	Up to four PCIe 4.0 x8
Ethernet hard IP	IEEE 802.3-compliant Clause 49 physical coding sublayer (PCS)	IEEE 802.3-compliant Clause 49 or Clause 107 PCS	
	Up to six 10 Gigabit Ethernet (GbE) media access control (MAC)	Up to six 10 or 25 GbE MAC	Up to sixteen 10 or 25 GbE MAC
	Supports IEEE 1588 Precision Time Protocol (PTP), Auto Negotiation and Link Training (AN/LT)		
Forward Error Correction (FEC)	IEEE 802.3 Clause 74 Firecode FEC	IEEE 802.3 Clause 74 Firecode FEC	
USB 3.1 hard IP(2)	One channel with USB 3.1 controller in HPS block		

Related Information

- Intel Agilex® 5 FPGAs and SoCs Device Overview
- Intel Agilex® 5 FPGAs and SoCs Device Data Sheet
- Intel Agilex® 5 Device Family Pin Connection Guidelines

(1) Supported for -4S speed grade (VCC=0.8V) devices only.

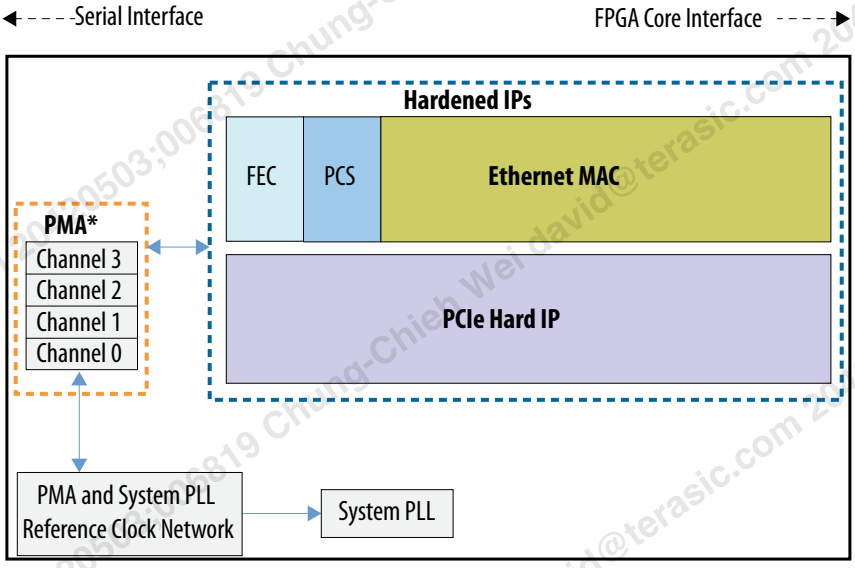
(2) Devices with GTS transceiver and HPS only.

2. GTS Transceiver Architecture

2.1. Building Blocks

A GTS transceiver bank consists of four PMA channels, hardened IPs (FEC, PCS, PCIe, and Ethernet MAC), a system PLL, and clock networks (for reference clock and datapath clock).

Figure 1. High-Level Block Diagram of a GTS Transceiver Bank



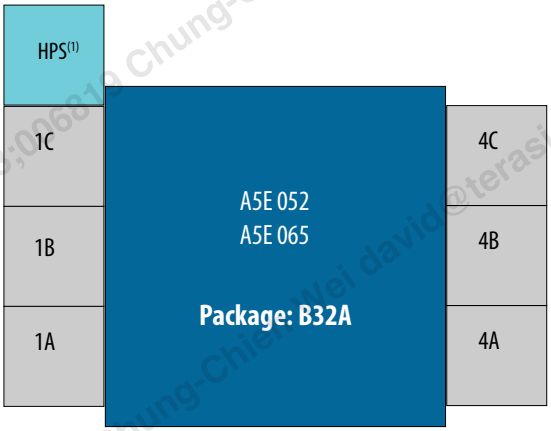
*Includes a control unit for PMA in the GTS transceiver bank

The number of GTS transceiver banks varies depending on device density and package variants. Refer to Intel Agilex® 5 FPGAs and SoCs Family Plan for details on the GTS transceiver count.

Refer to the following figures for the respective GTS transceiver bank layout. In devices with options for smaller packages, some GTS transceiver banks are downbonded and not available for use, except for the system PLL that remains available for use to clock the FPGA core logic.

Figure 2. GTS Transceiver Bank Layout for E-Series FPGAs with 24 GTS Transceivers

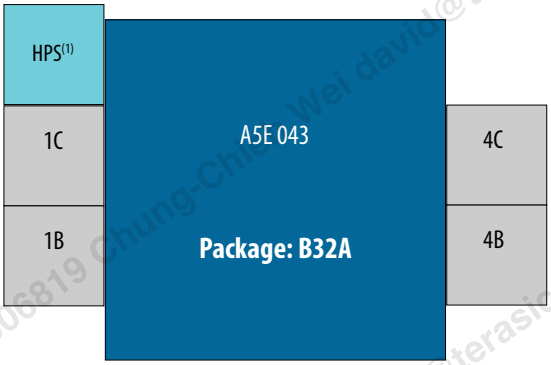
Applicable to Device Group A and Device Group B



(1) Only available in devices with GTS transceiver and HPS.

Figure 3. GTS Transceiver Bank Layout for E-Series FPGAs with 16 GTS Transceivers

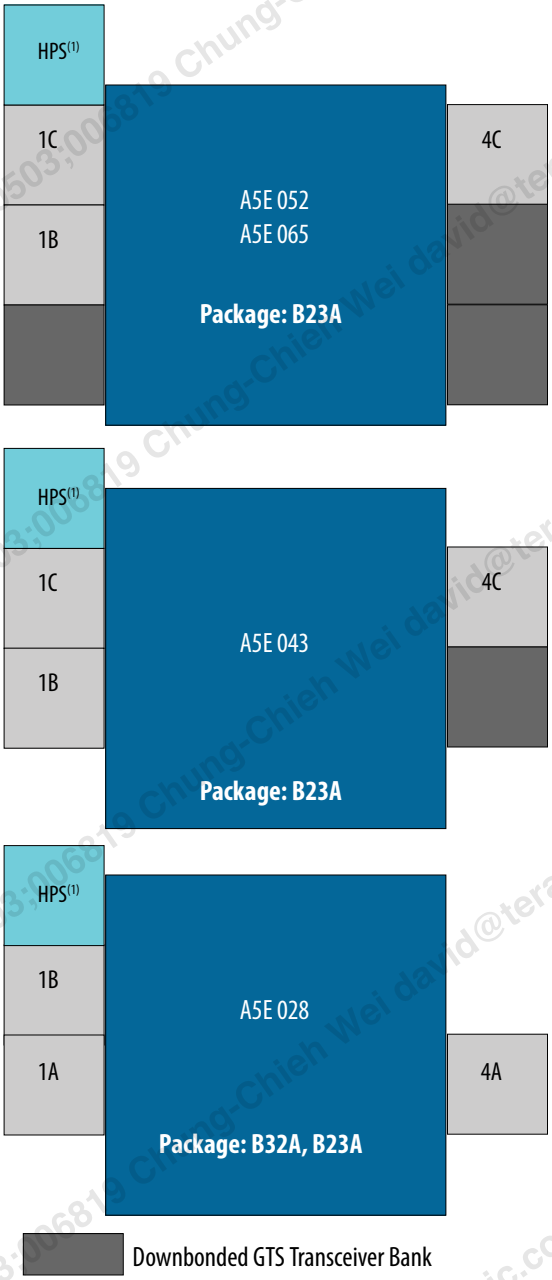
Applicable to Device Group A and Device Group B



(1) Only available in devices with GTS transceiver and HPS.

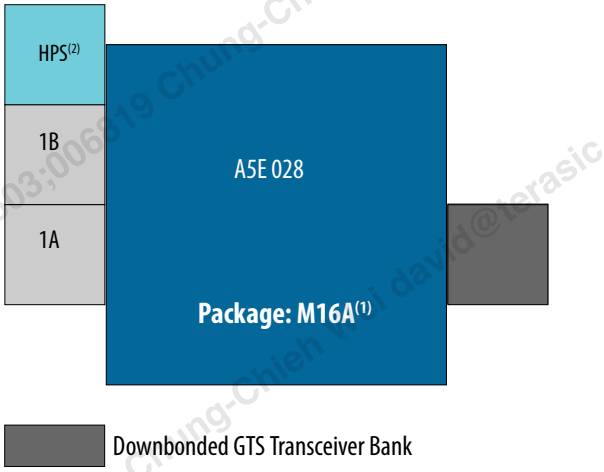
Figure 4. GTS Transceiver Bank Layout for E-Series FPGAs with 12 GTS Transceivers

Applicable to Device Group A and Device Group B



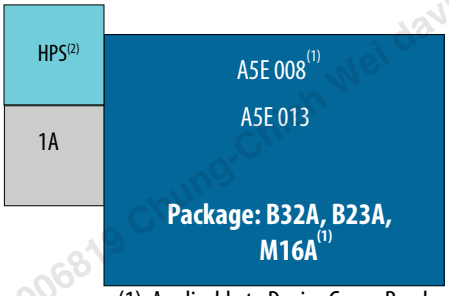
(1) Only available in devices with GTS transceiver and HPS.

Figure 5. GTS Transceiver Bank Layout for E-Series FPGAs with 8 GTS Transceivers



(1) Applicable to Device Group B only.
(2) Only available in devices with GTS transceiver and HPS.

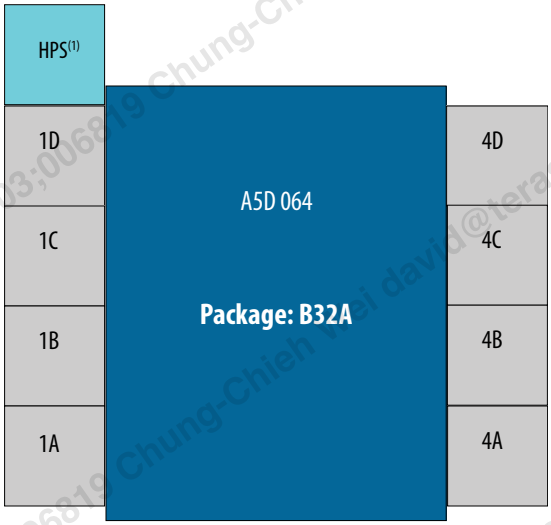
Figure 6. GTS Transceiver Bank Layout for E-Series FPGAs with 4 GTS Transceivers



(1) Applicable to Device Group B only
(2) Only available in devices with GTS transceiver and HPS.

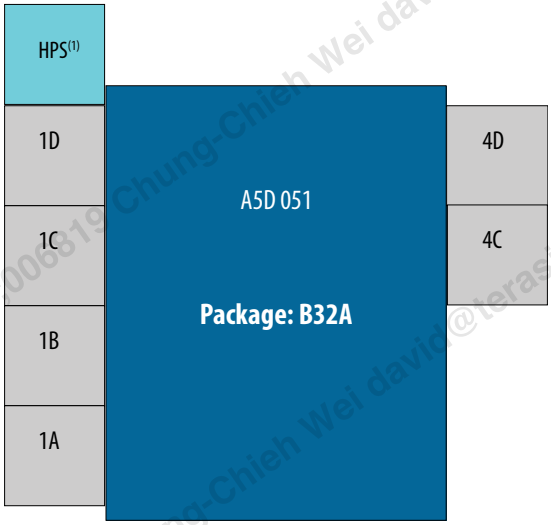
The following figures show the different packages and GTS transceiver combinations for the D-Series FPGAs.

Figure 7. GTS Transceiver Bank Layout for D-Series FPGAs with 32 GTS Transceivers



(1) Only available in devices with GTS transceiver and HPS.

Figure 8. GTS Transceiver Bank Layout for D-Series FPGAs with 24 GTS Transceivers

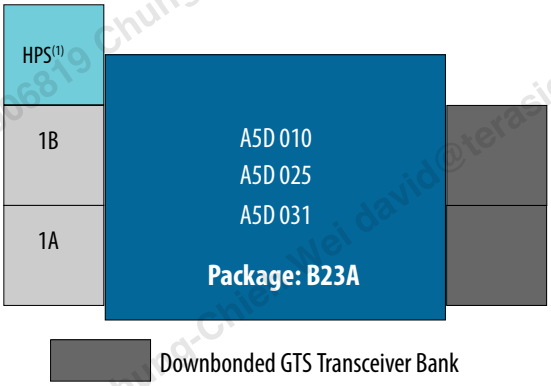


(1) Only available in devices with GTS transceiver and HPS.

Figure 9. GTS Transceiver Bank Layout for D-Series FPGAs with 16 GTS Transceivers

(1) Only available in devices with GTS transceiver and HPS.

Figure 10. GTS Transceiver Bank Layout for D-Series FPGAs with 8 GTS Transceivers



(1) Only available in devices with GTS transceiver and HPS.

The following table shows the hard IP configurations supported by the PMA for enabling various interface protocols.

Table 2. Hard IP Configurations Supported with PMA

Configuration	PCIe Hard IP	MAC	PCS	FEC	PMA	Example Protocols
Hardened PCIe IP	Yes	No	No	No	Yes	PCIe
Hardened Ethernet IP	No	Yes	Yes	Optional	Yes	10G/25G Ethernet
Hardened USB 3.1 IP(3)	No	No	No	No	Yes	USB3.1

continued...

(3) The hardened USB 3.1 IP controller resides in the HPS block, and is supported for devices with GTS transceiver and HPS only. Refer to the Intel Agilex 5 Hard Processor System Technical Reference Manual for implementation details of USB3.1.

Configuration	PCIe Hard IP	MAC	PCS	FEC	PMA	Example Protocols
PCS Direct	No	No	Yes	Optional	Yes	CPRI (64B/66B), FlexE, OTN
FEC Direct	No	No	No	Yes	Yes	Fibre Channel 16G
PMA Direct	No	No	No	No	Yes	Custom Interface, CPRI (8B/10B), HDMI, SDI, DisplayPort, JESD204B/C SATA, GPON, Fibre Channel, Interlaken

2.1.1. PMA

Each PMA channel offers exceptional signal integrity capability for NRZ transmission across various supported protocols and proprietary serial interfaces. The PMA includes transmit and receive buffers, serializer/deserializer, built-in debug features, individual TX PLL, and RX CDR PLL for independent channel operations. The PMA also has a control unit which is a firmware wrapper for RX adaptation, AN/LT for Ethernet and on-die instrumentation (ODI) functions. The independent transmitter and receiver supports combining simplex protocols implementations into a channel.

2.1.2. FEC

Each PMA channel can connect to an IEEE 802.3 compliant FEC that supports Firecode FEC (2112, 2080) or Reed Solomon FEC (528, 514) FEC modes.

2.1.3. PCS

Each PMA channel supports connecting to an IEEE 802.3 compliant Clause 49 or Clause 107 PCS layer with 64b/66b encoding and decoding, data scrambling, block alignment and gearboxing functions. The PCS block enables Ethernet 10G/25G, CPRI (64b/66b encoded), and OTN implementations.

2.1.4. Ethernet MAC

The Ethernet MAC together with PCS and the optional FEC offer a hardened protocol stack for Ethernet 10/25G implementations. The MAC supports 1588 PTP and AN/LT features.

2.1.5. PCI Express* Hard IP

Each GTS transceiver bank comprises a hardened protocol stack for PCI Express* controller in Endpoint, Root Port and TLP Bypass modes.

The PCI Express Hard IP is capable of PCI Express 1.0 to 3.0 or PCI Express 4.0 operating modes in x1, x2,x4, and up to x8 configurations.

Note: PCI Express 1.0 and 2.0 are supported via link down-training.

The key features that are supported are:

- Precision Time Measurement (PTM)
- FPGA core configuration via PCI Express link (CvP)
- Virtual I/O Device (VirtIO)

2.1.6. PLL and Clock Networks

The GTS transceiver bank has two types of highly configurable clock distribution networks that support the TX PLL in each PMA channel and system PLL:

- Reference clock – connects physical reference clock pins to TX PLL, RX CDR PLL, and system PLL.
- Datapath clock – drives PMA and hardened IPs from options of TX PLL (in PMA clocking mode) and system PLL (in system PLL clocking mode).

The GTS transceiver bank also supports:

- Multiple lane aggregation into a single link with bonding configuration that minimizes skew (from clock, reset, and interface synchronization) across the bonded lanes.
- Up to two clock pins in each GTS transceiver bank as receiver recovered clock outputs to provide a dedicated clock path to external clock cleaner (for example for CPRI applications).

2.1.7. Avalon Memory-Mapped Interface

Each GTS transceiver channel has an Avalon® memory-mapped interface for accessing the control and status registers (CSRs) for GTS transceiver building blocks except PCIe Hard IP. The PCIe Hard IP has a PCIe sideband streaming interface to access the CSR registers.

The Avalon memory-mapped interface also enables access to dynamically reconfigure a subset of PMA channels and hard IP blocks to operate in different modes.

2.2. Channel Placement Rules and Design Requirements

The GTS transceiver bank supports flexible channel placement that maximizes the utilization for a wide variety of protocol implementation by following the rules and requirements in this section.

2.2.1. Hard IP Rules

When planning for channel placement, follow the location requirements in the table below based on the required hard IP configuration in your design.

Table 3. Channel Placement Location Requirement for Supported Hard IP Configurations

Hard IP Configuration	Channel Placement Requirement
Hardened PCIe IP	Fixed locations as shown in Channel Placement for Hardened PCIe IP Configurations Across GTS Transceiver Banks
Hardened Ethernet IP	CH3 and CH2[4] in every GTS transceiver bank as shown in Channel Placement for Hardened Ethernet IP Configuration in Every GTS Transceiver Bank
Hardened USB3.1 IP	CH2 or CH1 in GTS transceiver bank directly adjacent to the HPS[5] block as shown in Channel Placement for Hardened USB3.1 IP Configuration in One GTS Transceiver Bank Directly Adjacent to the HPS Block. Refer to the Intel Agilex 5 Hard Processor System Technical Reference Manual for implementation details of USB3.1.
PCS Direct[6]	Any channel in a GTS transceiver bank, except for lane aggregation requiring bonding where location is as shown in Channel Placement for PMA Direct Configuration for Lane Aggregation Requiring Bonding
FEC Direct[6]	
PMA Direct	

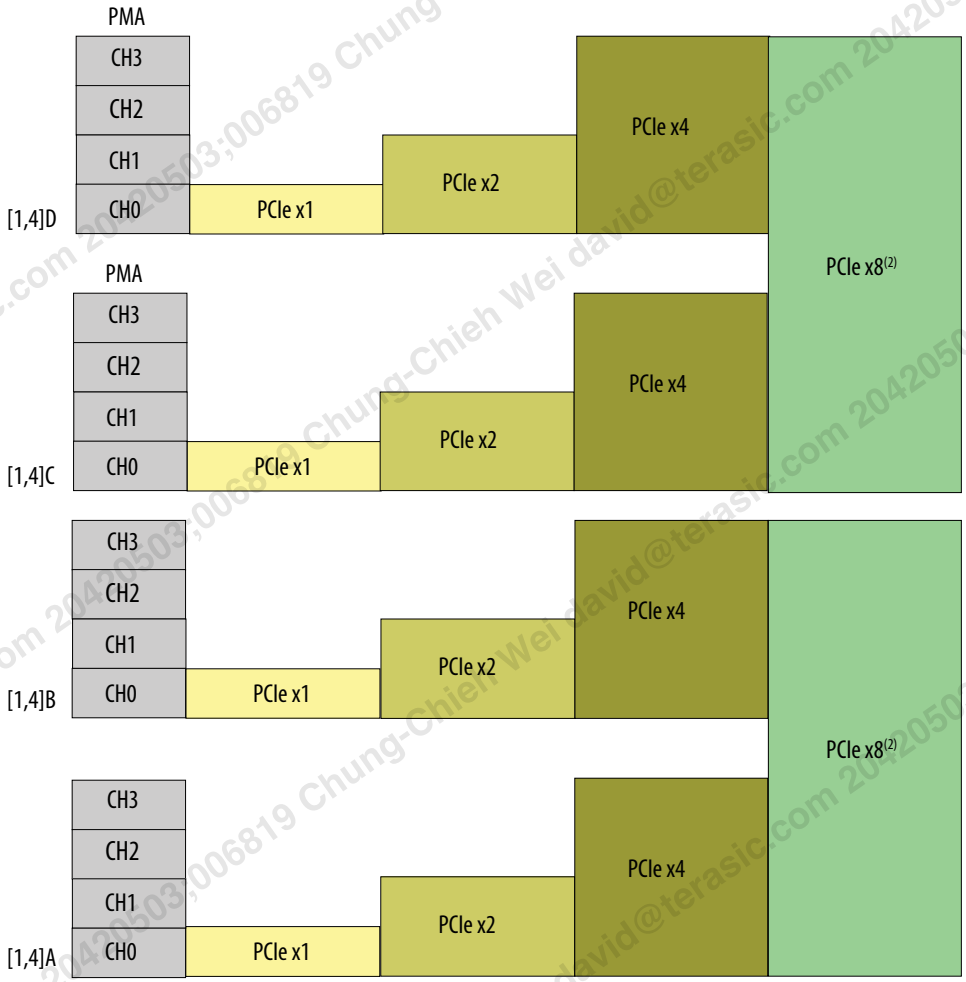
The figures below show the fixed location placement requirement for various configurations of supported hardened protocol IPs. The PMA channel that supports a particular configuration is shown in the same row as the hardened IP location or configuration.

(4) For D-series only.

(5) Devices with GTS transceiver and HPS only.

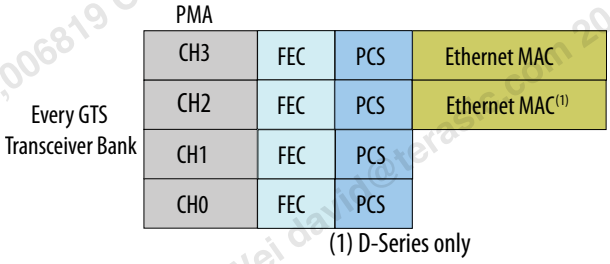
(6) Bonding not supported.

Figure 11. Channel Placement for Hardened PCIe IP Configurations Across GTS Transceiver Banks(1)



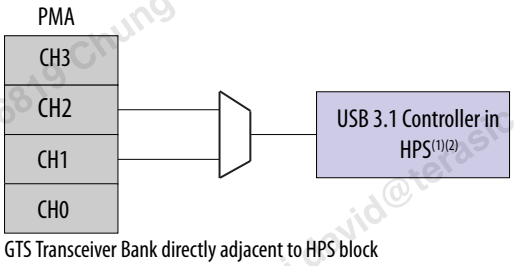
(1) Bank availability varies per device density and packages.
(2) D-Series only

Figure 12. Channel Placement for Hardened Ethernet IP Configuration in Every GTS Transceiver Bank



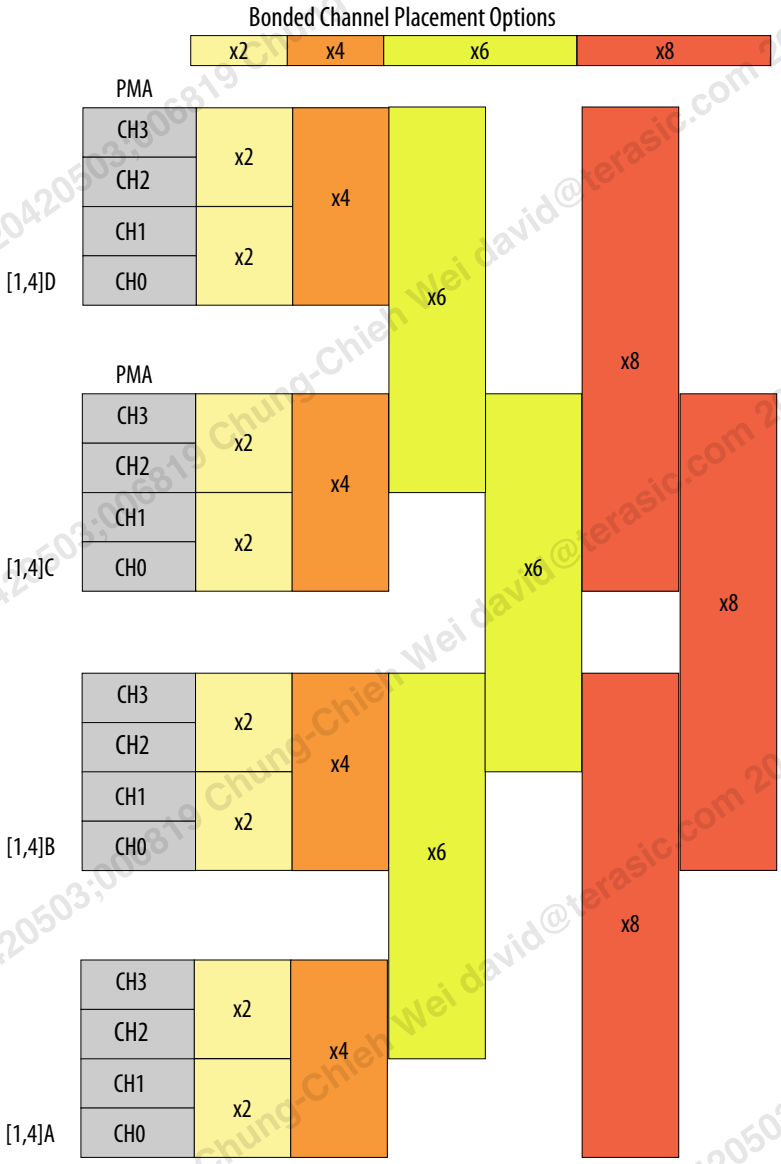
(1) D-Series only

Figure 13. Channel Placement for Hardened USB3.1 IP Configuration in GTS Transceiver Bank Directly Adjacent to the HPS Block



(1) Only supports 1 PMA channel (CH1 or CH2) to USB 3.1 Controller in HPS.
(2) Devices with transceiver and HPS only.

Figure 14. Channel Placement for PMA Direct Configuration for Lane Aggregation Requiring Bonding(1)



(1) Bank availability varies per device density and packages.

2.2.2. Use Scenario Rules

There are design rules that you must follow for the use scenarios described in the table below:

Table 4. Design Rules Associated for Specific Use Scenarios

Use Scenarios	Design Rules
Multiple interfaces within a GTS transceiver bank using the FEC block[7]	All FEC enabled interfaces within a GTS transceiver bank must be clocked by the same system PLL.
Independent simplex TX and simplex RX interfaces placed in same channel (dual-simplex mode)	Both TX and RX interfaces: • If not using PMA clocking, must be clocked by the same System PLL. • FEC is not supported in dual-simplex mode. • Must share one Avalon memory-mapped interface to access the channel. Enable arbiter logic if you need independent Avalon memory-mapped interface access to the simplex TX and simplex RX interfaces placed in same channel.
Multiple lanes bonded to single link	When bonded channels use a system PLL, they must all use the same system PLL.

2.2.3. Unused PMA Rules

To save power, you can power down unused GTS transceiver banks that you do not plan to use in the future. Connect the PMA power supplies of the unused banks to ground to power them down. This is supported only in either of the following conditions:

1. All GTS transceiver banks on the same side are unused.
2. The specified one, two or four unused GTS transceiver banks as listed in Supported Specific GTS Transceiver Bank Power Down table below.

Note: Device and package combinations not listed in the following table only support case 1 above.

Table 5. Supported Specific GTS Transceiver Bank Power Down

Device	GTS Transceiver Banks that Support Power Down		
	B32A Package	B23A Package	M16A Package[8]
A5E 028	1A	1A	1A
A5E 042	–	1B	[9]
A5E 052	1A, 4A, or both	1B	[9]
A5E 065	1A, 4A, or both	1B	[9]
A5D 010	1A, 4A, or both	1A	[9]
A5D 025	1A, 4A, or both	1A	[9]
A5D 031	1A, 4A, or both	1A	[9]
A5D 051	1A and 1B	[9]	[9]
A5D 064	1A and 1B, 4A and 4B, or all 1A, 1B, 4A and 4B	[9]	[9]

(7) Refer to FEC Architecture for details of the FEC core.

(8) Device group B only.

(9) No package combination for this device.

You must adhere to the following to implement power down for the unused banks in the current Intel® Quartus® Prime Pro Edition software:

- Connect the PMA power supplies of the unused banks to ground.
- Do not assign TX/RX/REFCLK pins to the powered down banks.
- Check and ensure no TX/RX/REFCLK pins are auto-assigned to the powered down banks.

The power estimation is currently higher as all unused GTS banks are assumed to be powered. The power down setting is planned to be supported in a future release of the Intel Quartus Prime Pro Edition software.

You must preserve currently unused PMA channels that you plan to use in the future as described in the table below.

Table 6. Steps to Preserve Unused PMA Channels

Unused PMA Channel Scenario	Steps to Preserve PMA Channel
Unused GTS transceiver bank	Do not power down the bank. Connect PMA power supply to GTS transceiver bank.
Unused PMA channel not instantiated in design	No action required as channels are preserved by default.
Unused PMA channel instantiated in design	Hold TX and RX PMA in reset by asserting the `i_tx_reset` and `i_rx_reset` ports of the GTS PMA/FEC Direct PHY Intel FPGA IP.

For TX PMA channels in operation, ensure that the data pattern is toggling. Do not send data patterns consisting of all zeros or all ones. If the reference clock is disconnected during operation, hold the TX and RX PMA channels in reset before disconnecting the reference clock.

2.2.4. General Design Requirement

In a design with GTS transceivers, you must provide an external clock to OSC_CLK_1 pin.

2.3. PMA Architecture

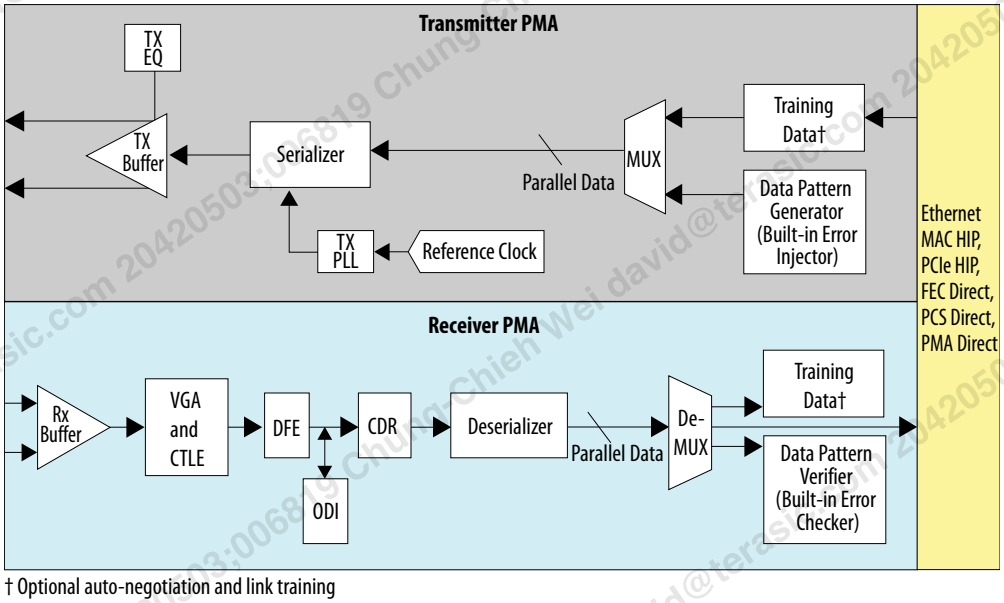
The PMA supports the data rates as shown in the following table.

Table 7. Supported PMA Data Widths and Date Rates

PMA Width	Modulation	E-Series FPGAs Device Group B Data Rates (Gbps)	E-Series/D-Series FPGAs Device Group A Data Rates (Gbps)
		PMA/System PLL Clocking (1 GHz Max)	
8	NRZ	1-8	1-8
10	NRZ	1-10	1-10
16	NRZ	1-16	1-16
20	NRZ	1-17.16	1-20
32	NRZ	1-17.16	1-28.1

The PMA block diagram is shown in the following figure.

Figure 15. PMA Block Diagram



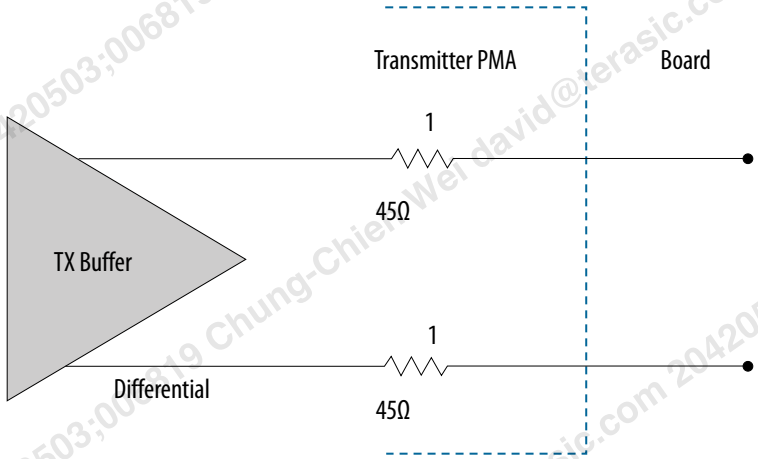
† Optional auto-negotiation and link training

2.3.1. Transmitter PMA Architecture

2.3.1.1. Transmitter Buffer

A simplified transmitter buffer termination scheme is shown in the following figure.

Figure 16. Simplified TX Buffer Termination



1. $Z_{TX-DIFF-DC}$ transmitter buffer output differential DC impedance is 90 Ω; 45 Ω single ended.

The transmitter buffer can be programmed to support the taps listed in the following table.

Table 8. Transmitter PMA Equalizer Parameters

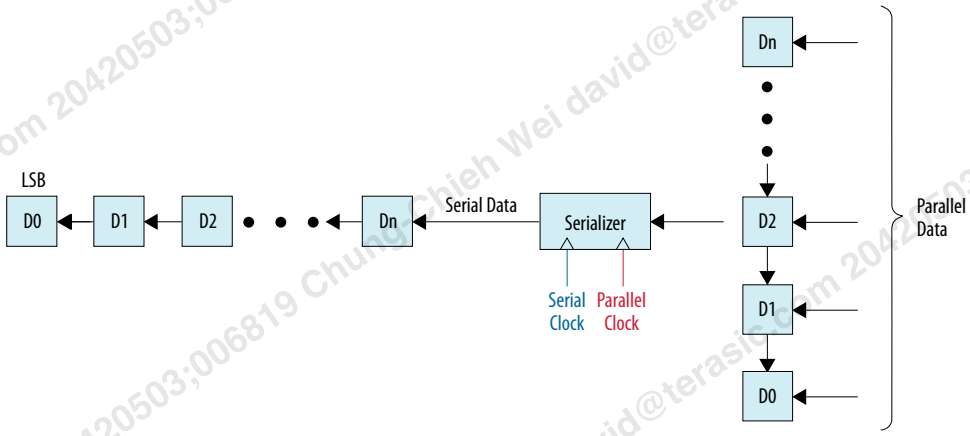
Parameter	Cursor	Rule		Increment and Decrement Size
		Minimum	Maximum	
pre_tap_2	C_{-2}	0	+7	1.0
pre_tap_1	C_{-1}	0	+15	1.0
main_tap	C_0(10)	0	+55	1.0
post_tap_1	C_{+1}	0	+19	1.0

2.3.1.2. Serializer

The serializer converts the parallel data into a serial data stream.

The serializer supports the following serialization factors: 8, 10, 16, 20, 32. The serializer is hard coded to transmit the least significant bit first. The serializer block diagram is shown in the following figure.

Figure 17. Serializer



2.3.1.3. Data Pattern Generator and Verifier

The PMA supports a data pattern generator and verifier.

The PMA supports a built-in transmitter data pattern generator for transmit characterization. The pattern and size are programmable. The supported programmable PRBS patterns are shown below:

- PRBS31
- PRBS28
- PRBS23
- PRBS15
- PRBS13
- PRBS10

(10) C_0 = main_tap + 1 - pre_tap_1 -pre_tap_2 - post_tap_1

- PRBS9
- PRBS7
- SSPR
- SSPR1

The data pattern verifier contains a receiver built-in self test (BIST) bit error checker. The receiver can check standard data patterns for link verification applications by enabling the PRBS mode in both the receiver link and a compatible transmitter link connected by a common transmission path or loopback.

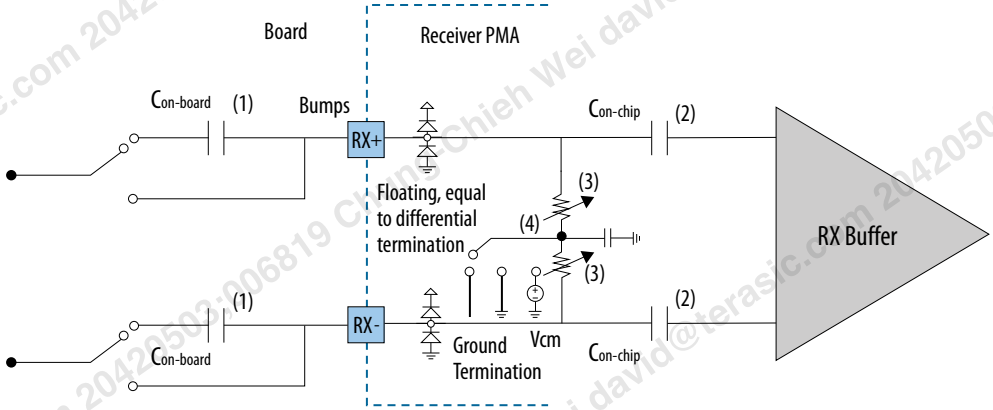
Note: Selecting the PRBS generator and verifier is planned for support through the GTS PMA/FEC Direct PHY Intel FPGA IP in a future release of the Intel Quartus Prime Pro Edition software.

2.3.2. Receiver PMA Architecture

2.3.2.1. Receiver Buffer and Equalizer

The receiver analog front end is shown in the following figure.

Figure 18. Simplified Receiver Analog Front End



The various capacitors and resistors for the receiver analog front end are described below:

1. You can implement on board AC coupling capacitors, $C_{on\text{-}board}$, based on applicable standards.
2. $C_{on\text{-}chip}$, on-chip AC coupling capacitor is 1pF. It is always on and is only bypassed in SDI mode.
3. $R_{DIFF\text{-}DC}$, DC differential receive impedance is programmable to 85Ω or 100Ω.
4. When you implement on-board AC coupling capacitors you must set $V_{RX\text{-}CM\text{-}DC}$ to ground termination. When it is DC coupled and no on-board AC coupling capacitors are implemented, $V_{RX\text{-}CM\text{-}DC}$, receiver input DC common-mode voltage (non-SDI mode) at the bumps must be:
   a. Smaller than 700mV, if squelch detect is not used.
   b. Must be between 200mV to 300mV, if squelch detect is used.

   V_{cm} is set to 700mV automatically if you use SDI mode.

Note: Setting V_{cm} is planned to be supported in a future release of the GTS PMA/FEC Direct PHY Intel FPGA IP in the Intel Quartus Prime Pro Edition software.

The receiver buffer receives serial data from input pins and feeds it to the CDR block and deserializer. To optimize the bit error rate (BER) for optimum performance, receiver equalization is self-triggered, requires no input, and is independent of system initial conditions.

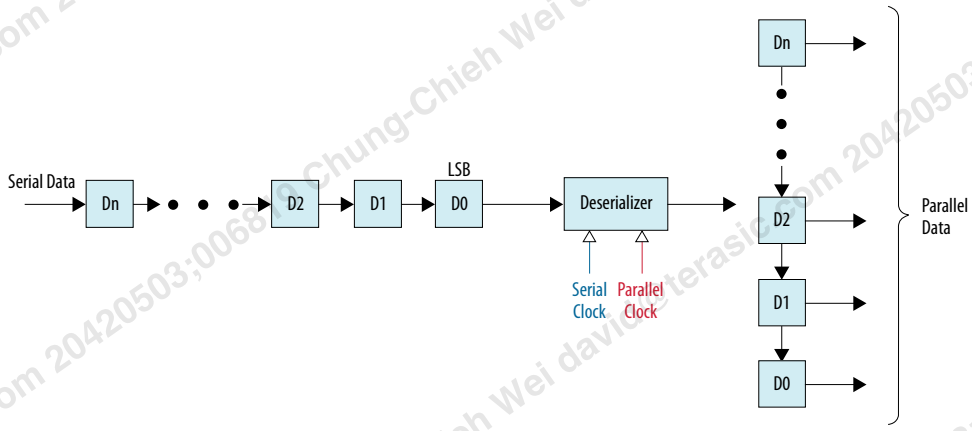
2.3.2.1.1. Control Unit

The control unit is a firmware wrapper that takes over RX adaptation from the PMA. It also handles auto-negotiation and link training for Ethernet. The control unit can also provide some support for on-die instrumentation (ODI). It is not used when PMA is operating in PCIe mode.

2.3.2.2. Deserializer

The deserializer clocks in serial input data from the receiver buffer using the high speed serial recovered clock, and deserializes the data using the low-speed parallel recovered clock. The deserializer forwards the deserialized data to the receiver PCS, FEC, PCIe HIP, USB HIP or FPGA core. The deserializer supports the following deserialization factors: 8, 10, 16, 20, 32.

Figure 19. Deserializer



2.3.2.3. CDR Block

Clocking resources in the receiver enable the clock data recovery feature. When locked, the CDR extracts the clock from the received data. The CDR supports automatic and manual lock mode.

2.3.2.4. PMA Tuning

The PMA supports adaptive VGA, Continuous Time Linear Equalization (CTLE) and Decision Feedback Equalization (DFE). Auto adaptation of the VGA, CTLE and DFE equalizes a broad range of channels across temperature variation. The PMA also supports manual tuning. The PMA uses auto adaptation by default unless you bypass the auto adaptation.

Note: The steps to enable manual tuning is planned for support in a future release of the Intel Quartus Prime Pro Edition software.

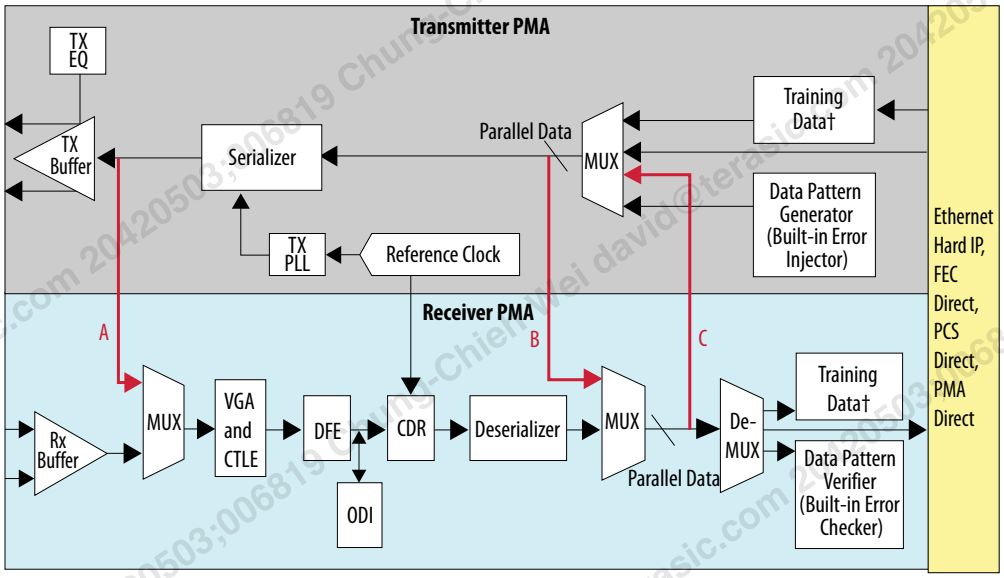
2.3.3. PMA Loopback Modes

The PMA channel includes serial and parallel loopback paths. These loopback paths provide support for different debugging options and configurations. Available loopback paths are labeled in the following figure and described below:

- A: PMA-transmitter-to-receiver internal serial loopback:
  - Loops back the transmitter pre-driver differential I/O signals to the receiver equalizer.
- B: PMA-transmitter-to-receiver parallel loopback:
  - Parallel loopback from the PMA transmit lane 32 bit data ports to the receive lane 32 bit data ports.
- C: PMA-receiver-to-transmitter reverse parallel loopback:
  - Parallel loopback from the PMA receive lane 32 bit data ports to the transmit lane 32 bit data ports. This loopback enables the transmitter buffer to transmit data fed directly from the CDR recovered data.

Note: Only parallel loopback, mode (B), is supported in the current Intel Quartus Prime Pro Edition software release. Internal serial loopback, mode (A), and reverse parallel loopback, mode (C), are planned to be supported in a future release.

Figure 20. PMA Loopback Modes



† Optional auto-negotiation and link training

2.4. FEC Architecture

The IEEE 803.2 compliant FEC block is located between the PCS and PMA interface. The FEC block has four instances of clause 74 Firecode FEC (2112, 2080) that are optional and you can bypass them.

Table 9. Supported FEC Modes and Compliance Specifications

FEC Mode	Specification	FEC Compliance Specification	Example Protocols
Firecode(11)	IEEE	IEEE 802.3 BASE-R Firecode (CL 74)	10GbE

Note: If your configuration has multiple interfaces in one FEC block, the block must be clocked by the same system PLL and requires custom cadence. Refer to Datapath Clock Cadences for details.

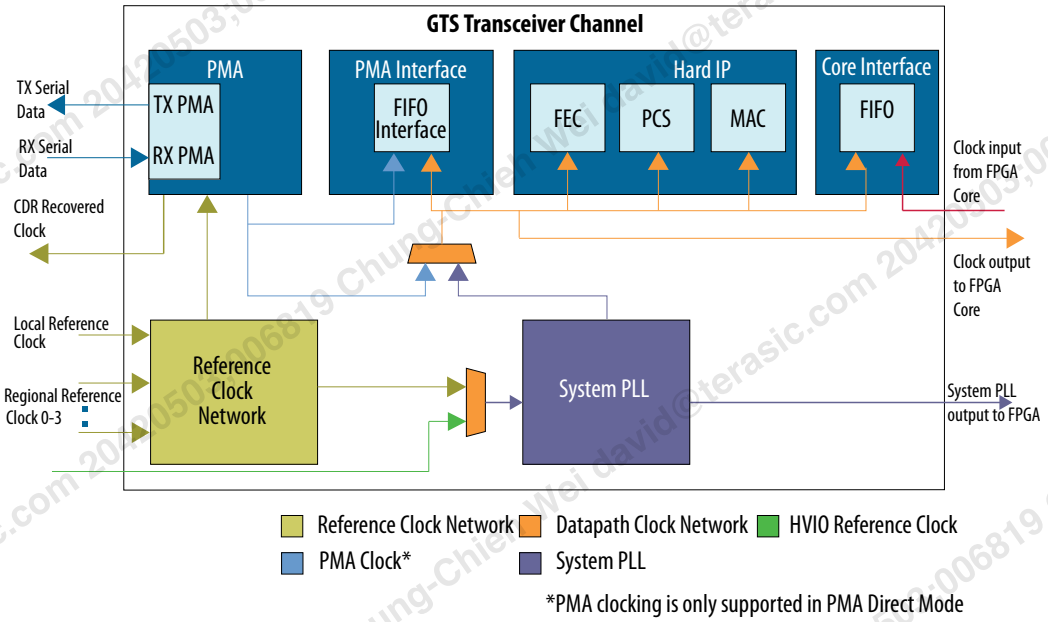
2.5. Clock Architecture

The Intel Agilex 5 FPGA GTS transceivers have two types of clock networks:

- Reference clock network
- Datapath clock network

The following figure is a simplified block diagram of the clock networks.

Figure 21. Clock Network



Both the system PLL and the PMA clock get their reference clock from the reference clock network. The datapath clock network is accessible by all enabled digital blocks and is driven by a clock from either the PMA or the system PLL. PMA clocking includes the use of a Transmit PLL (TX PLL) or Clock Data Recovery (CDR) for the transmit and receive clocks respectively.

(11) Firecode FEC 25GbE mode is supported in D-Series and E-Series Device Group A. Firecode FEC 25GbE mode is not supported in E-Series Device Group B.

2.5.1. Reference Clock Network

There are two types of clock lines in the reference clock network; local and regional reference clock lines. For every GTS transceiver bank, there are two independent differential reference clock input pins.

The first is the local reference clock, which can reach any channel in the same GTS transceiver bank and the system PLL. This local reference clock pin is bi-directional, except on devices with a single GTS transceiver bank on a side, where it serves as an input only reference clock pin. You can configure this bi-directional pin as an output for the CDR recovered clock from any of the four channels in the GTS transceiver bank. In one of the transceiver banks of the device, there is an additional output only pin of this type. The bank location where this output only CDR recovered clock resides varies across the different device variants. You can find the bank location of the CDR recovered clock output pins in the device pinout files.

The second available input reference clock pin is the regional reference clock. This reference clock can be used to feed all the channels and system PLLs in the two closest GTS transceiver banks above or below its own bank. For example, in a four GTS transceiver bank device, the regional reference clock pin in bank 0 can be used in bank 1 and bank 2 as well. However bank 3 would be out of reach of the regional reference clock pin in bank 0.

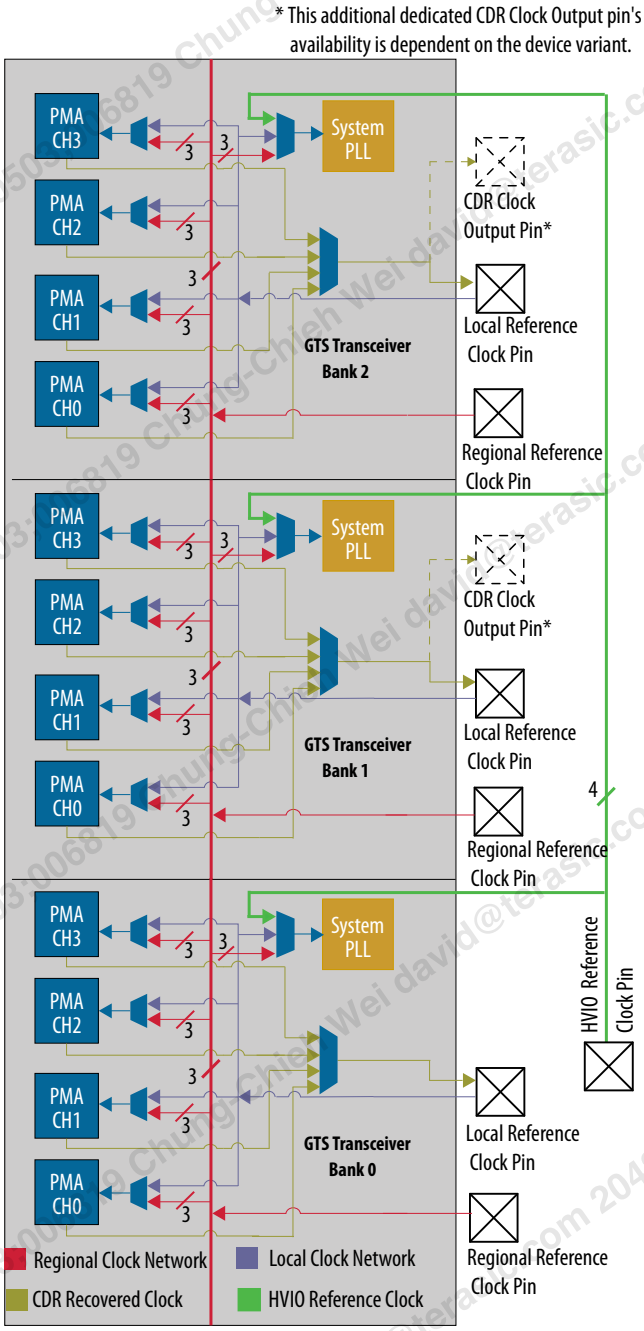
Additionally, the system PLL can also get its input reference clock from outside the GTS transceiver bank. Adjacent to the GTS transceiver bank, there is a high voltage IO (HVIO) bank. There are 4 single-ended input reference clock pins in the HVIO bank that can be used by the system PLL as a secondary option for the reference clock input source.

Table 10. Reference Clock Source Comparison

Description	Local Reference Clock	Regional Reference Clock	HVIO Reference Clock
Used by	PMA and system PLL	PMA and system PLL	System PLL
Reach	All PMA channels and system PLL within a GTS transceiver bank	All PMA channels and system PLL in the two closest GTS transceiver banks above or below its own bank.	All system PLLs on the same side of device
Recovered clock output	Available	Not available	Not available
Input/Output	Bi-directional, either: • Input reference clock • Output recovered clock	Input only	Input only
Pin location	GTS Transceiver bank	GTS Transceiver bank	HVIO5B or HVIO6A bank
Signal type	Differential	Differential	Single ended

The following figure shows the reference clock network and reference clock input pins for GTS transceiver banks.

Figure 22. Reference Clock Network for Devices with Multiple GTS Transceiver Banks on a Side



The PMA (TX PLL and CDR) reference clock sources are:

- Local reference clock input
- Regional reference clock input (from same GTS transceiver bank or from the neighboring GTS transceiver banks along the same side of the device)

System PLL reference clock sources are:

- Local reference clock input
- Regional reference clock input (from same GTS transceiver bank or from the neighboring GTS transceiver banks along the same side of the device)
- Dual purpose single ended pins (up to four) from HVIO5B or HVIO6A banks.

The CDR in any one of the four PMA channels within the GTS transceiver bank can send the recovered clock externally, for example to feed an external jitter cleaner. The output recovered clock is fed externally through the bi-directional local reference clock pin. The input reference clock is not available when this pin is being used for the recovered clock. In this scenario, the reference clock can come from any of the regional reference clock lines.

In addition to the bidirectional local reference clock pin, there is one additional dedicated CDR clock out pin in one of the GTS transceiver banks. You can use this pin to drive out the recovered clock. You must use this pin if you are planning to migrate to the smallest device variant in the future. This is because devices with a single GTS transceiver bank on a side have a slightly different clock network. Refer to Single GTS Transceiver Bank Device for more information.

Related Information

- General Purpose I/Os in Intel Agilex 5 FPGAs and SoCs
- Intel Agilex 5 A5EC065B ES Device Pinouts
- Intel Agilex 5 A5ED065B ES Device Pinouts
- TX and RX Reference Clock and Clock Output Interface Signals on page 57
- AN 979: Intel Agilex 5 E-Series Device Migration Guidelines

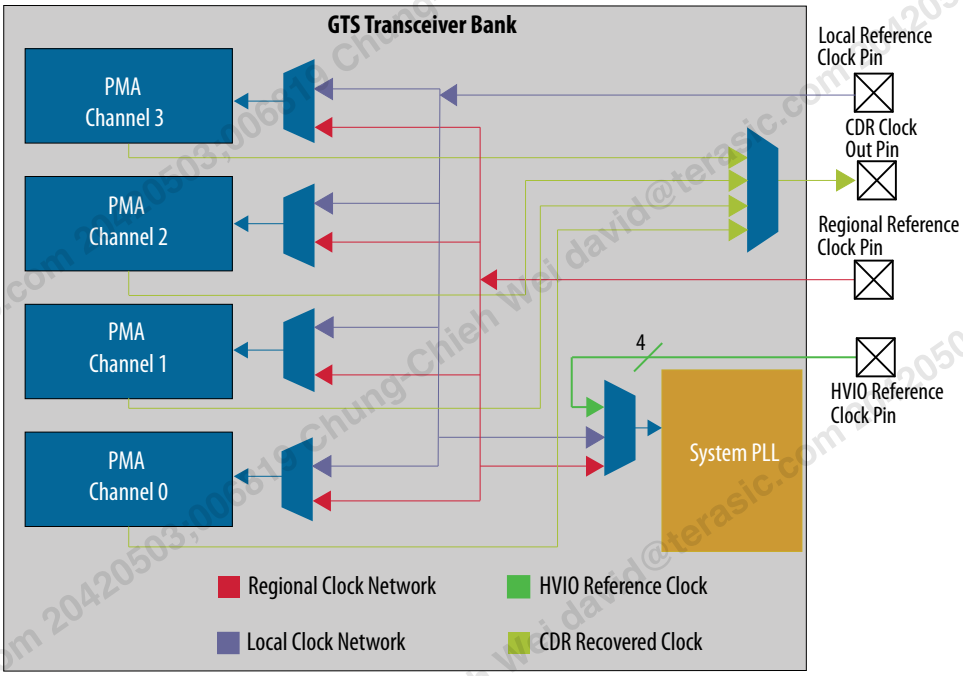
2.5.1.1. Single GTS Transceiver Bank Device

There are some devices that have a single GTS transceiver bank in the device, and some devices with multiple GTS transceiver banks but have a single GTS transceiver bank on the right side of the device. For devices where there is only one GTS transceiver bank on a side, the reference clock network differs slightly. For this single transceiver bank, the local reference clock pin is input only. However, there is a second output-only recovered clock pin. The following devices are single GTS transceiver bank devices on either the left or right side:

- A5E 028[12]
- A5E 008
- A5E 013

(12) Only applicable to the single GTS transceiver bank on the right side of the device. The multiple GTS transceiver banks on the left side of the device use the reference clock network of multiple GTS transceiver bank devices.

Figure 23. Reference Clock Network for Devices with a Single GTS Transceiver Bank on a Side



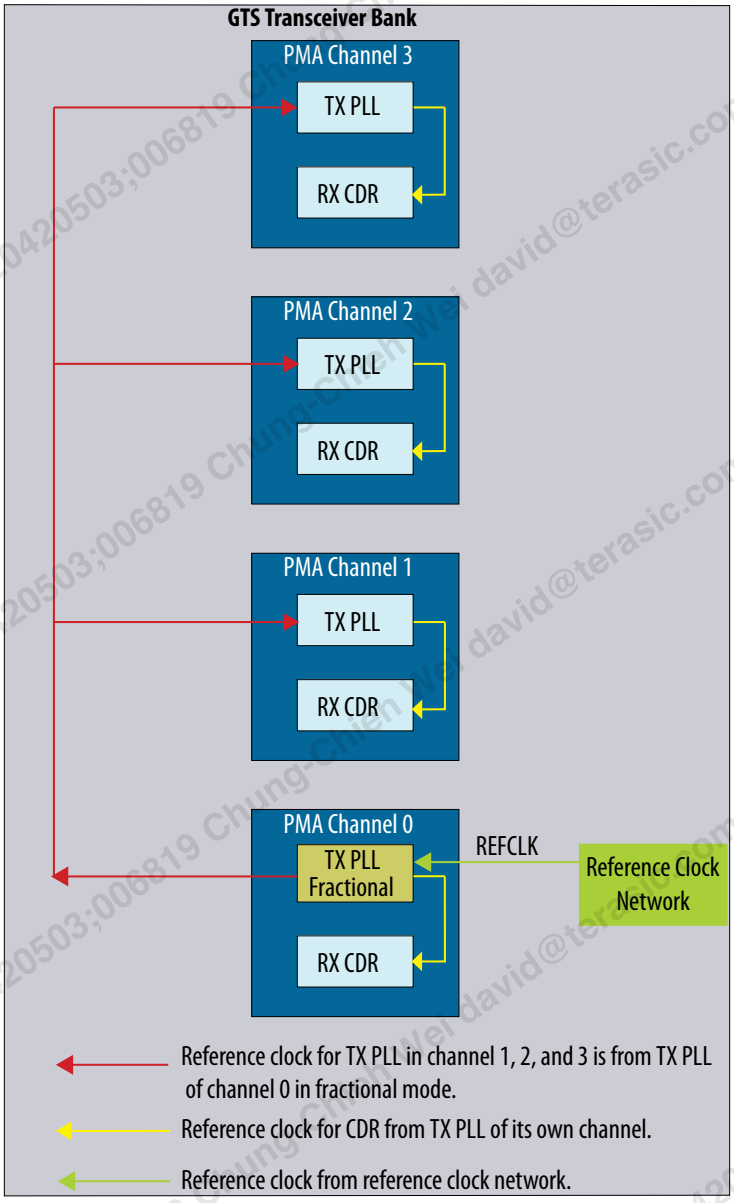
2.5.1.2. PMA Primary PLL Configuration

The Primary PLL configuration can be used for OTN applications, in which you have a group of channels that are configured in fractional mode and you want to change the fractional value dynamically.

In the primary PLL configuration the reference clock source for the local CDR and the TX PLL of other channels can be sourced from the PMA channel 0 TX PLL in the same GTS transceiver bank. The TX PLL of channel 0 must be in fractional mode and acts as the reference clock to the TX PLL of other channels. The clock source for the RX CDR of channels is provided by the TX PLL of its own channel. Refer to the following figure for more information.

Primary PLL configuration works only in a four channel configuration, where PMA channel 0 is always providing the reference clock to the TX PLL of the other channels. The primary PLL configuration is limited within a GTS transceiver bank and cannot extend beyond the GTS transceiver bank.

Figure 24. Primary PLL Configuration



Refer to *Primary PLL configuration* in PMA Support for Fractional Mode table for more information about the implementation details.

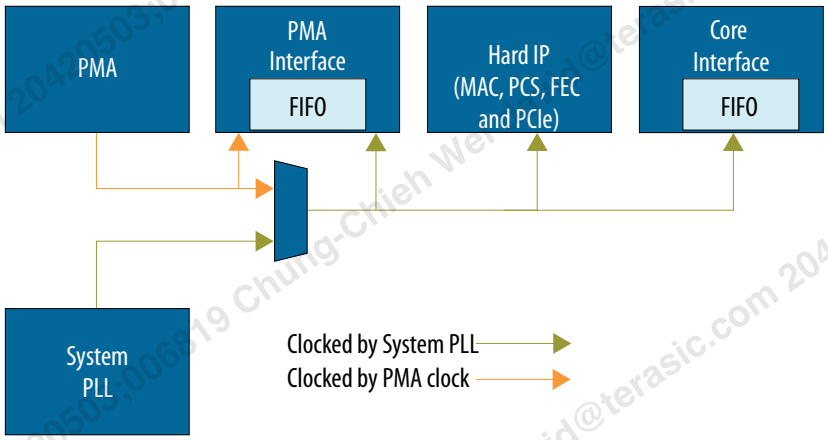
2.5.2. Datapath Clock Network

There are 2 datapath clocking modes, the system PLL clocking mode and the PMA clocking mode.

For the system PLL clocking mode, where the datapath is clocked by the system PLL:

- The system PLL clocking mode must be used for PCIe, Ethernet, PCS direct and FEC direct modes.
- The PMA direct mode can also be clocked by system PLL.
- System PLL clocking must be used for dynamic reconfiguration.

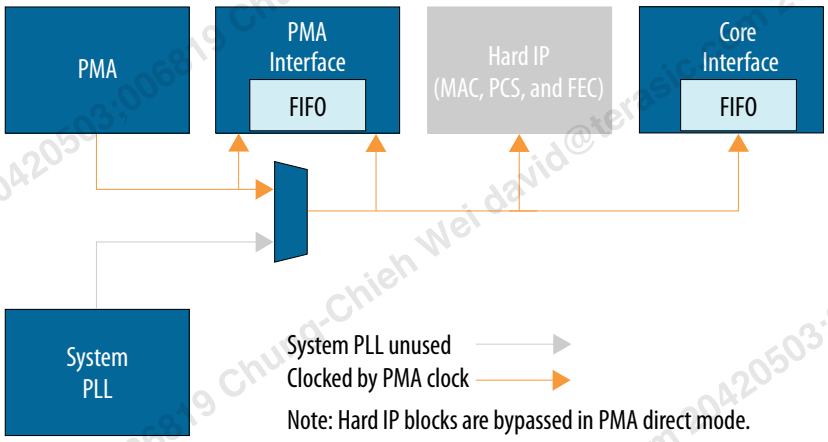
Figure 25. System PLL Clocking Mode



For the PMA clocking mode, which can only be used for PMA direct mode:

- TX datapath is clocked by the TX PLL
- RX datapath is clocked by the CDR

Figure 26. PMA Clocking Mode



Refer to Supported PMA Data Widths and Date Rates for supported data rates.

2.5.3. System PLL

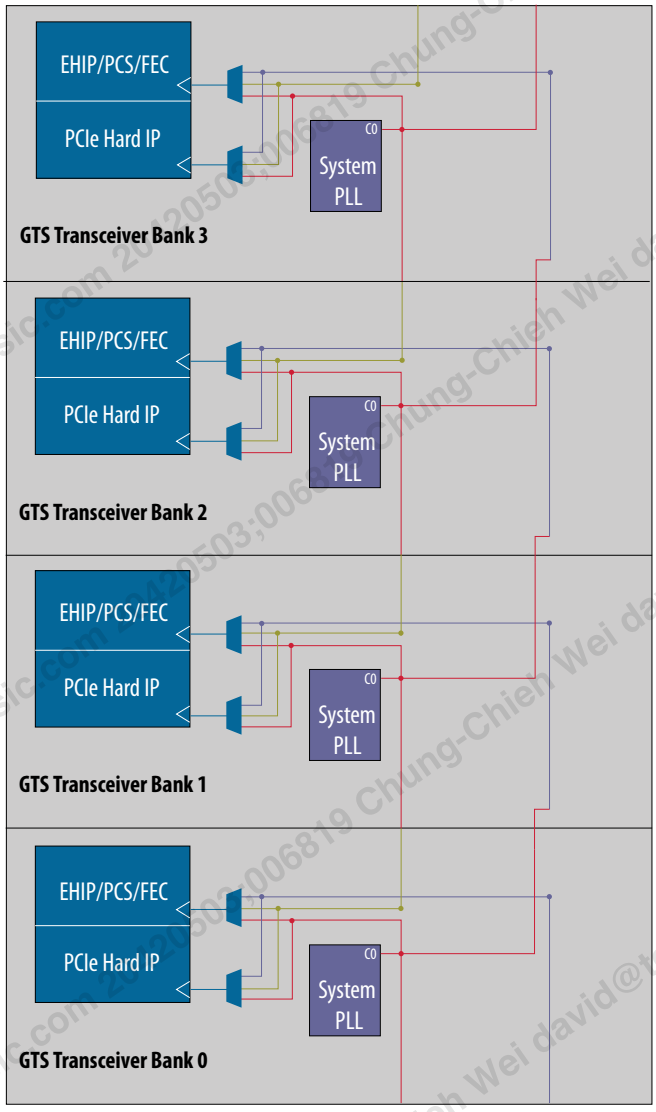
Each GTS transceiver bank has one system PLL. The system PLL is the primary clock source for hard IP blocks (Ethernet MAC, PCS, FEC and PCIe) and the core interface which bridges the FPGA core and the GTS transceivers.

The system PLL has one output (C0) to feed those blocks. When you use the system PLL clocking mode, the hard IP blocks are not clocked by the PMA clock. The system PLL can also drive hard IPs in the transceiver banks immediately above it or below it. The system PLL can also be used to clock the PMA direct mode.

You must instantiate and configure the system PLL using the GTS System PLL Clocks Intel FPGA IP. For more information, refer to Implementing the GTS System PLL Clock Intel FPGA IP.

Each system PLL can use either of the local reference clock or regional reference clock in the GTS transceiver bank, or the regional reference clocks coming from other GTS transceiver banks. It can also get the reference clock from four HVIO pins located in the adjacent HVIO bank.

Figure 27. System PLL clock network



Different interface protocols operating at different line rates can share a system PLL, except for PCIe. When multiple interface protocols share a system PLL, the protocol with the highest line rate determines the system PLL frequency, and the protocols with the lower line rates must be overclocked. The exact cadence is based on the clock; refer to Datapath Clock Cadences for the details.

The following table shows an example where four interfaces share a system PLL where:

- The system PLL is configured for the 25GbE datapath interface (the highest line rate of all four interfaces)
- The three lower line rate datapath interfaces are overclocked and need custom cadence

Table 11. Example of a Single System PLL Shared Between Interfaces

Interface Protocol	Line Rate (Gbps)	PMA Width	PMA Clock Frequency (MHz): Line Rate / PMA width	System PLL Frequency (MHz)	System PLL Output-to-Core Frequency (MHz)	Datapath Clock Frequency
25 GbE	25.78125	32	805.67	805.67	402.83	Same as the PMA clock frequency
10 GbE Soft MAC	10.3125	32	322.26	805.67	402.83	Over-clocked to the PMA clock frequency
10.1376 Gbps CPRI	10.318	32	316.81	805.67	402.83	Over-clocked to the PMA clock frequency
9.8 Gbps CPRI	9.8304	20	491.52	805.67	402.83	Over-clocked to the PMA clock frequency

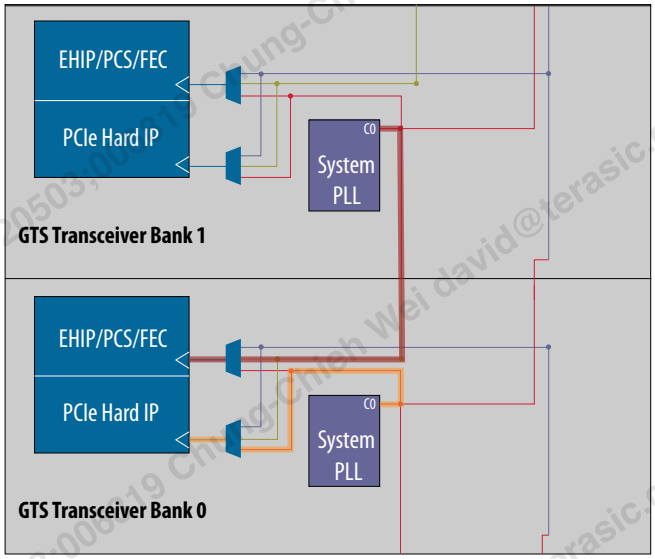
The system PLL must be shared by channels using FEC if they belong to the same transceiver bank.

2.5.3.1. Running PCIe and Non-PCIe Protocols in a GTS Transceiver Bank

When you configure a GTS transceiver bank to run PCIe and non-PCIe protocols, you must use two system PLLs. Since there is only one system PLL in a GTS transceiver bank, a second or third system PLL can be used from the neighboring GTS transceiver banks. For devices with a single transceiver bank, refer to HVIO PLL as System PLL for additional information.

For example, if channels 0 and 1 in GTS transceiver bank 0 are configured for PCIe, and channels 2 and 3 are configured for Ethernet, then the system PLL in GTS transceiver bank 0 can be configured for PCIe, and the system PLL from GTS transceiver bank 1 can be configured for Ethernet. The following figure shows the system PLL connections for the example described above.

Figure 28. Example Using System PLL From a Different GTS Transceiver Bank



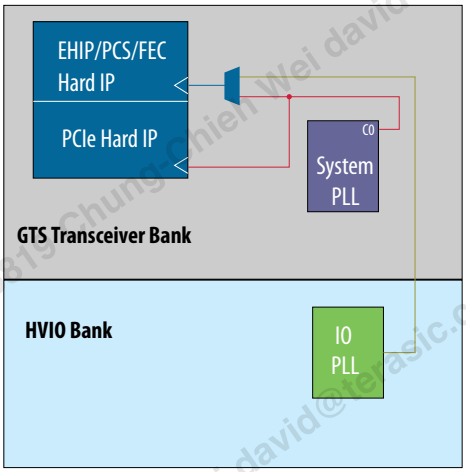
2.5.3.2. I/O PLLs in HVIO Bank as System PLL

For certain devices, there is only one GTS transceiver bank and therefore only one system PLL is available. For these devices, you can use the I/O PLL in the adjacent HVIO bank as a second system PLL. This enables you to configure multiple protocols operating at different frequencies.

The following devices have only one GTS transceiver bank and one system PLL:

- A5E 008
- A5E 013

Figure 29. I/O PLL in HVIO Bank Usage



As the I/O PLL is different from the system PLL, you have to instantiate the I/O PLL using the *IOPLL Intel FPGA IP* instead of the *GTS System PLL Clocks Intel FPGA IP*. Refer to the Intel Agilex 5 Clocking and PLL User Guide for more information.

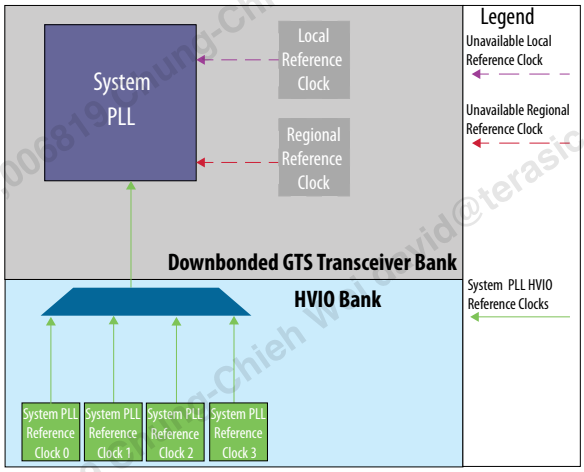
Related Information

- Building Blocks on page 6
- I/O PLLs in Intel Agilex 5 FPGAs and SoCs

2.5.3.3. System PLL with HVIO Reference Clock

In the scenario where the GTS transceiver bank is downbonded as shown in Building Blocks, the system PLL can still be used for the FPGA core. However, the system PLL does not have access to the GTS transceiver's local or regional reference clocks. The reference clock for the system PLL has to come from the single-ended HVIO pin in the HVIO bank adjacent to the GTS transceiver bank as shown in the following figure.

Figure 30. System PLL with HVIO Reference Clock from Adjacent HVIO Bank



2.5.4. Datapath Clock Cadences

The read and write frequency of the PMA FIFO interface determines if you need a standard or custom cadence.

- Standard cadence: Use if the read and write frequencies of the PMA FIFO interface are the same with 0 ppm frequency delta.
- Custom cadence: Use if the read and write frequencies of the PMA FIFO interface have different frequencies or have the same frequency but with a frequency delta of greater than 0 ppm.

Table 12. Supported Datapath Clock Frequencies and Cadences by Datapath Clocking Mode

Datapath Clocking Mode	Configuration	Datapath Clock Frequency	Cadence
PMA clocking mode	PMA Direct	Datapath clock frequency = PMA clock frequency	Use the standard cadence on the TX and RX (data is valid at every clock edge).(13)

continued...

(13) The TX PMA and TX digital blocks use a PMA clock derived from the local clock. The RX PMA and RX digital blocks run on a recovered clock (the link partner clock).

Datapath Clocking Mode	Configuration	Datapath Clock Frequency	Cadence
(maximum 1 GHz)		PMA clock frequency = line rate/PMA width	
System PLL clocking mode (maximum 1 GHz)	PMA Direct	Use Case A: Chip-to-chip applications where the PMA channel and link partner share the same reference clock Datapath clock frequency ≥ (system PLL output frequency)min where (system PLL output frequency)min = PMA clock frequency	If (system PLL output frequency = PMA clock frequency and Δppm = 0), use the standard cadence on the TX and RX (data is valid at every clock edge). Otherwise, use custom cadence.(14), (15)
		Use Case B: Applications where the PMA channel and link partner do not share the same reference clock Datapath clock frequency ≥ (system PLL output frequency)min where (system PLL output frequency)min = (maximum ppm(16) ÷ 1000000 + 1) × PMA clock frequency	
System PLL clocking mode (maximum 1 GHz)	Other configurations with FEC, PCS, and MAC	Datapath clock frequency ≥ (system PLL output frequency)min where (system PLL output frequency)min = PMA clock frequency For example, for 10GbE-1, use ≥ 322.265625 MHz; and for 25GbE-1, use ≥ 805.6640625 MHz.	If (system PLL output frequency = PMA clock frequency), use the standard cadence on the TX and RX (data is valid at every 32 of 33 or 34 clock cycles). Otherwise, use custom cadence.(17)

Refer to Supported PMA Data Widths and Date Rates for supported data rates.

PMA Direct 28.1 Gbps PMA Clocking Mode Example

- All blocks between the PMA interface and core interface FIFO are bypassed and all enabled blocks run on the PMA clock.
- On the transmitter, the TX PMA interface FIFO is clocked by the TX PMA clock on both sides.
- On the receiver, the RX PMA interface FIFO is clocked by the RX recovered clock on both sides.
- Use the standard cadence. Data on the TX and RX is valid at every clock edge of the PMA clock.

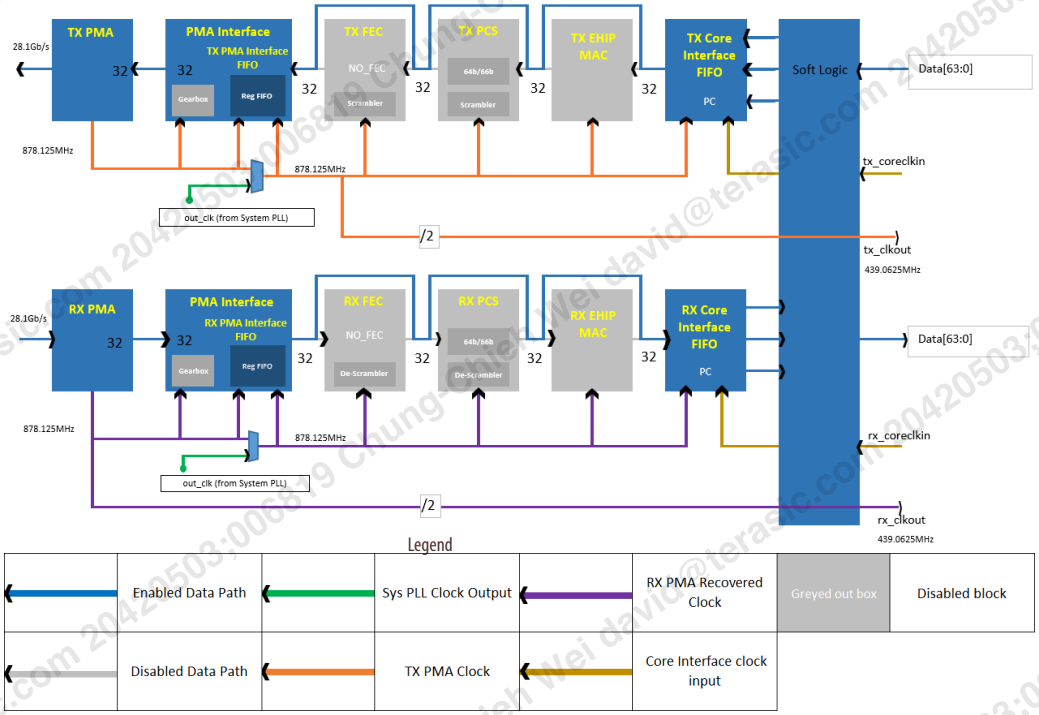
(14) Use Case A: Standard cadence can be used only when the TX PMA reference clock, system PLL reference clock, and link partner TX reference clock are coming from same clock source (with a 0 ppm frequency delta).

(15) Use Case B: The system PLL frequency must be overclocked to compensate for a frequency delta of greater than 0 ppm between the TX PMA reference clock, system PLL reference clock, and link partner TX reference clock.

(16) maximum ppm = maximum Δppm ÷ 2
maximum Δppm = max(Δppm between the link partner TX (the recovered clock on the local RX) and system PLL, Δppm between the system PLL and TX PMA)

(17) The data path clock is already overclocked compared to the PMA clock by approximately 3% because of PCS and FEC overhead. Therefore, a frequency delta of greater than 0 ppm between the TX PMA reference clock, system PLL reference clock, and link partner TX reference clock is allowed.

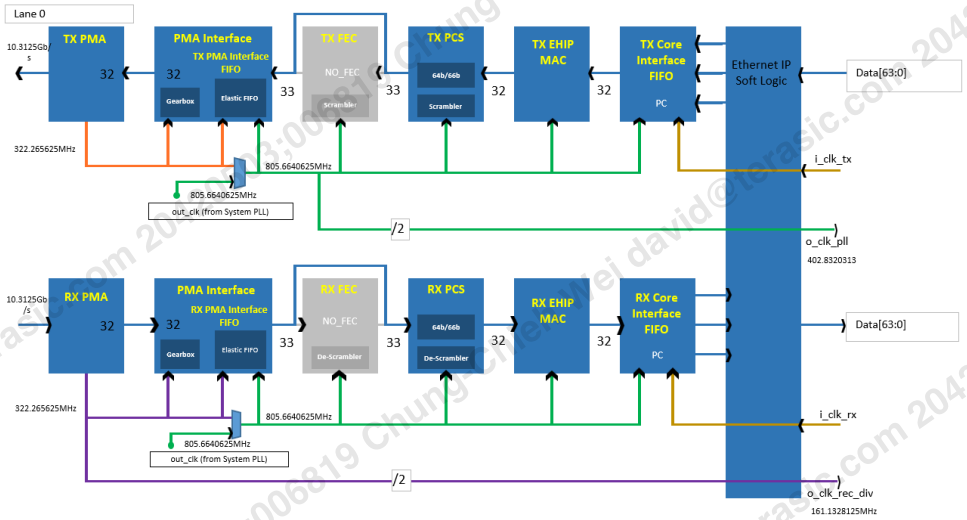
Figure 31. PMA Direct 28.1 Gbps PMA Clocking Mode Example



10 Gbps Ethernet with MAC and PCS Blocks Using Overclocked System Clocking Mode Example

- The blocks of the core interface FIFO, Ethernet hard IP MAC and PCS, and the PMA interface FIFO are clocked by the system PLL.
- On the transmitter, the TX PMA interface FIFO performs a clock transfer from the system PLL domain to the TX PMA clock domain.
- On the receiver, the RX PMA interface FIFO performs a clock transfer from the RX recovered clock domain to the system PLL domain.
- Because the system PLL clock frequency is faster than the PMA clock frequency, datapath clocking is overclocked. Therefore, you must use custom cadence.

Figure 32. 10 Gbps Ethernet with MAC and PCS Blocks Using Overclocked System Clocking Mode Example



2.6. Bonding and Deskew

2.6.1. Bonding Architecture

The objective of bonding is to minimize the TX lane-to-lane skew for multi-lane configurations to support protocols that have stringent skew requirements.

Bonding is applied for the following modes:

- PMA Direct

Supported channel bonding configurations are x2, x4, x6 and x8 as shown in Channel Placement Rules and Design Requirements. Bonding works with the following clocking modes:

Table 13. Bonding Clocking Modes

Modes	Clocking Modes
PMA Direct	PMA Clocking/System Clocking

For bonded configurations, only phase compensation mode of the core interface FIFO is supported.

2.6.2. TX Deskew Function

- When channels are bonded, skew between core interfaces is minimized to reduce the skew between the channels. TX deskew logic is turned on if there are more than one PMA channel. Deskew logic in the transmit direction aligns the core interface by generating internal deskew pulses every 16 or 32 clock cycles. These pulses are used to align the transmit parallel data before driving into the PMAs.

3. Implementing the GTS PMA/FEC Direct PHY Intel FPGA IP

The following chapters describe the implementation of GTS transceiver physical (PHY) layer IP, PLLs and clock networks. Refer to the chapters for implementation details of IP instantiation, connection, and simulation, and placement of the GTS transceivers.

Implementation of GTS PMA/FEC PHY designs involves instantiation and connection of the following required and optional Intel FPGA IPs that are available in the Intel Quartus Prime IP catalog:

- GTS PMA/FEC Direct PHY Intel FPGA IP (Required)
- GTS System PLL Clocks Intel FPGA IP (Required only if using system PLL clocking mode)
- GTS Reset Sequencer Intel FPGA IP (Required)

This user guide organizes the information into the following chapters describing the IP and implementation:

- Implementing the GTS PMA/FEC Direct PHY Intel FPGA IP — describes functions, parameters, and ports, bit mapping, core clocking, reset and bonding of the IP.
- Implementing the GTS System PLL Clocks Intel FPGA IP — describes the function, parameters, and ports of the IP.
- Implementing the GTS Reset Sequencer Intel FPGA IP — describes the function parameters and ports of the IP.
- GTS PMA/FEC Direct PHY Design Implementation — describes instantiation, connection, simulation and interface planning using an example design.

3.1. IP Overview

The GTS PMA/FEC Direct PHY Intel FPGA IP is for use in proprietary protocol configurations. The GTS PMA/FEC Direct PHY Intel FPGA IP enables access to the PMA Direct and FEC Direct modes.

The PMA Direct mode bypasses the MAC and FEC Hard IP blocks. You can configure the PMA interface and core interface FIFOs in the datapath into various modes, including elastic and phase compensation modes.

The FEC Direct mode bypasses the MAC and PCS Hard IP blocks. In this mode, the PMA interface and core interface FIFOs in the datapath are set to elastic and phase compensation modes, respectively.

The following figures show the PMA Direct datapath and FEC Direct datapaths with various clocking modes:

Figure 33. PMA Direct Mode with PMA Clocking

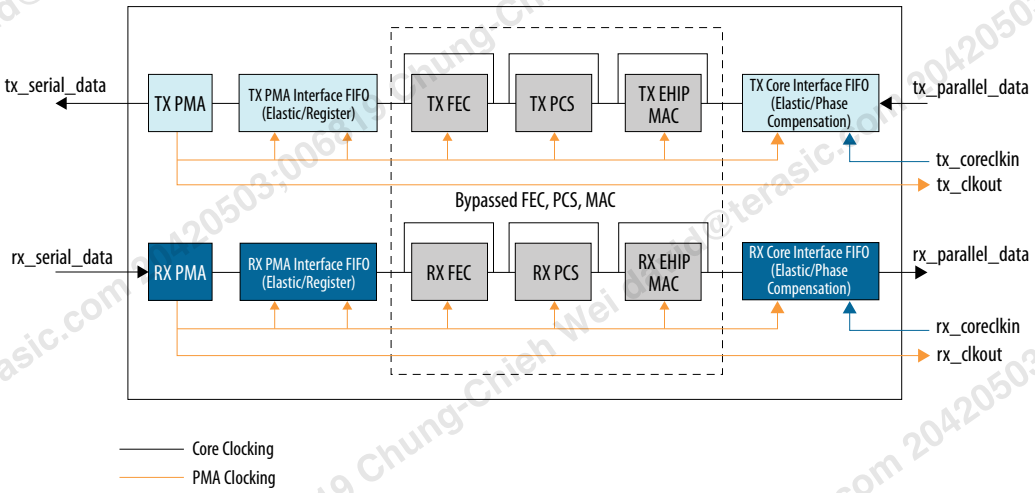


Figure 34. PMA Direct Mode with System Clocking

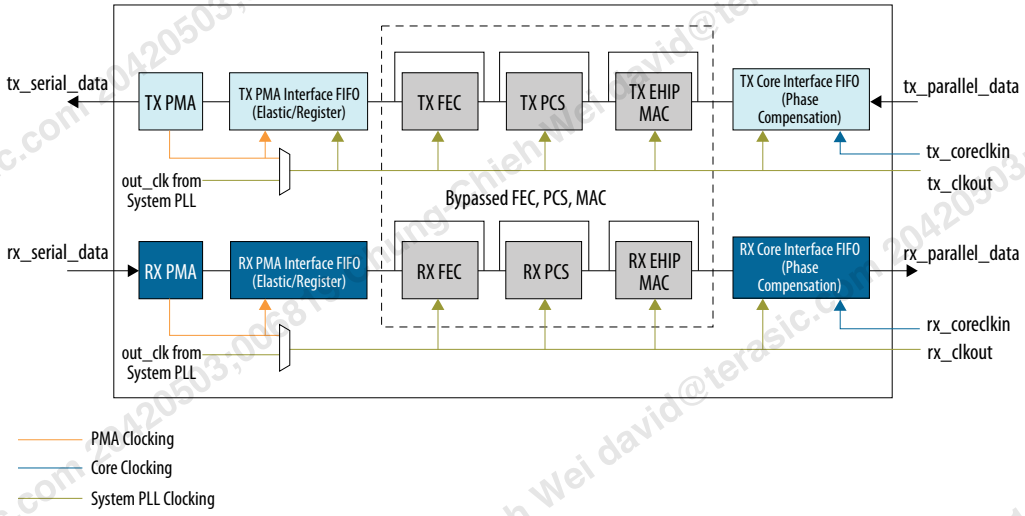
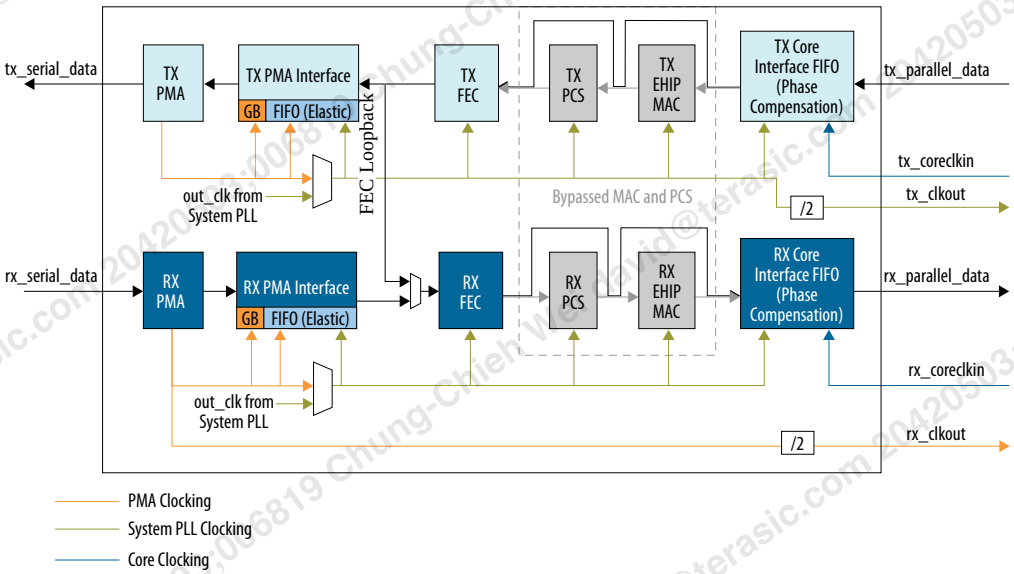


Figure 35. FEC Direct Mode with System PLL Clocking



You can use the PMA/FEC Direct PHY Intel FPGA IP to configure the datapath into the PMA and FEC direct modes. If you enable the FEC mode, the FEC block is enabled. The top-level file that generates with the IP instance includes all the available ports for your configuration. Use these ports to connect the GTS PMA/FEC Direct PHY Intel FPGA IP to other IP cores in your design, such as GTS System PLL Clock Intel FPGA IP, GTS Reset Sequencer Intel FPGA IP, TX and RX serial data pins, data generator and data checker soft IP.

3.1.1. PMA Direct Supported Modes

The GTS PMA/FEC Direct PHY Intel FPGA IP currently supports the following PMA Direct modes:

- NRZ modulation
- Duplex, TX simplex and RX simplex modes for both PMA clocking and system PLL clocking with 8, 10, 16, 20, and 32 data widths.
- Supports x2, x4, x6 and x8 bonding on the TX path
- Supports configurable FIFO modes: PMA interface FIFO (elastic and register modes) and core interface mode (phase compensation)

Table 14. PMA Direct Mode Support

Clocking Mode	Double Width/Single Width Mode[18]	PMA Interface Width	PMA Interface FIFO (TX/RX)	Core Interface FIFO (TX/RX)
System Clocking	DW	8, 10, 16 ,20,32	Elastic/Elastic	Phase Compensation/Phase Compensation
	SW	8, 10 ,16, 20, 32	Elastic/Elastic	Phase Compensation/Phase Compensation
PMA Clocking	DW	8, 10, 16, 20, 32	Register/Register	Phase Compensation/Phase Compensation
		8, 10, 16, 20, 32	Register/Register	Elastic/Phase Compensation
		8, 10, 16, 20, 32	Register/Register	Phase Compensation/Elastic
		8, 10, 16, 20, 32	Register/Register	Elastic/Elastic
	SW	8, 10, 16, 20, 32	Register/Register	Phase Compensation/Phase Compensation
		8, 10, 16, 20, 32	Register/Register	Elastic/Phase Compensation
		8, 10, 16, 20,32	Register/Register	Phase Compensation/Elastic
		8, 10, 16, 20, 32	Register/Register	Elastic/Elastic

For multiple lanes and TX deskew function, core interface FIFO must be set to phase compensation mode.

3.1.2. FEC Direct Supported Modes

The GTS PMA/FEC Direct PHY Intel FPGA IP supports the following for FEC Direct mode:

- IEEE 802.3 Firecode (2112,2080, CL 74)
  - Support for 10G and 4x10GbE
- Supports only the System PLL clocking mode

You can enable the FEC Direct mode in the IP parameter editor by turning on the **Enable FEC** option. The FEC Direct modes with FEC specifications are topology dependent to achieve different BER. FEC data to and from the PCS is 33 bits. On the PMA interface side, FEC data from and to the PMA interface is 33 bits wide. For designs that include FEC, the gearbox enables automatically. The gearbox option for Firecode FEC is 32:33.

[18] The Double width (DW) mode is when the **Enable TX/RX double width transfer** parameter in the GTS PMA/FEC Direct PHY Intel FPGA IP GUI is enabled. When it is enabled, you can clock the FPGA core logic with a half rate clock. Single width (SW) mode is when this parameter is not enabled.

Table 15. FEC Direct IP Configuration Mode Support

Clocking Mode	FEC Mode	Double Width/ Single Width[19]	PMA Interface Width	PMA Interface FIFO (TX/RX)	Core Interface FIFO (TX/RX)
System PLL Clocking	Firecode (2112,2080)	DW	32	Elastic/Elastic	Phase Compensation/ Phase Compensation

3.1.3. Unsupported PMA/FEC Modes

The GTS PMA/FEC Direct PHY Intel FPGA IP does not support the following PMA/FEC modes:

- No support for TX simplex and RX simplex mode for FEC Direct mode.
- Gearbox feature is disabled for PMA Direct mode.

3.2. Designing with the GTS PMA/FEC Direct PHY Intel FPGA IP

The GTS PMA/FEC Direct PHY Intel FPGA IP is the primary IP component for PMA and FEC direct usage. This IP provides direct access to the Intel Agilex 5 GTS PMA block features.

To customize and instantiate the IP for your protocol implementation, you specify parameter values for the GTS PMA/FEC Direct PHY Intel FPGA IP and generate the IP RTL and supporting files from the Intel Quartus Prime parameter editor. The top-level file generated with the IP instance includes all the available ports for your configuration.

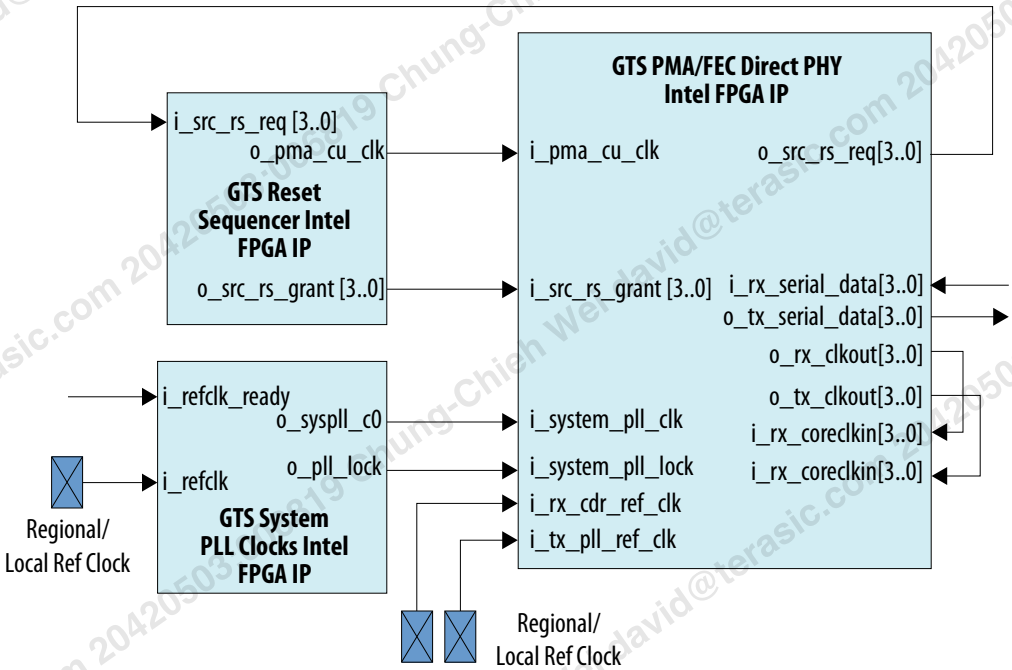
The GTS PMA/FEC Direct PHY Intel FPGA IP allows you to configure and support PMA and FEC direct modes with the following:

- Datapath Clocking mode, PMA mode, PMA data rate, PMA width
- TX datapath and RX datapath options settings (FIFO modes, TX PLL, RX CDR)
- FEC Options
- Avalon Memory Mapped Interface

The following figure shows the block diagram of the GTS PMA/FEC Direct PHY Intel FPGA IP connections to the relevant IP blocks. This is an example of the connections that you have to make for the system PLL clocking mode. If you are using the PMA clocking mode, you do not need to instantiate the GTS System PLL Clocks Intel FPGA IP.

(19) The Double width (DW) mode is when the **Enable TX/RX double width transfer** parameter in the GTS PMA/FEC Direct PHY Intel FPGA IP GUI is enabled. When it is enabled, you can clock the FPGA core logic with a half rate clock. Single width (SW) mode is when this parameter is not enabled.

Figure 36. GTS PMA/FEC Direct PHY Intel FPGA IP Connections



3.3. Configuring the GTS PMA/FEC Direct PHY Intel FPGA IP

Use the GTS PMA/FEC Direct PHY Intel FPGA IP in the Intel Quartus Prime Pro Edition software to configure the PMA PHY for your protocol implementation. To instantiate the IP, follow these steps:

1. To specify the target device family, click **Assignments** -> **Device**, and then select an Intel Agilex 5 device.
2. Click **Tools ➤ IP Catalog**, type PMA in the search field, and select the GTS PMA/FEC Direct PHY Intel FPGA IP (under Interface Protocol). The IP parameter editor opens.
3. In the parameter editor, specify the parameters to customize the GTS PMA/FEC Direct PHY Intel FPGA IP for your protocol implementation. Select one of the following PMA usage modes. The parameter editor guides your parameter value selections.
   - PMA Direct mode
   - FEC Direct mode
4. When parameterization is complete, click the **Generate** button, and then click the **Generate HDL** button. Your IP variation RTL and supporting files generate according to your specifications, and are added to your Intel Quartus Prime project. The top-level file generated with the IP instance includes all the available ports for your configuration. Use these ports to connect the GTS PMA/FEC Direct PHY Intel FPGA IP to other IP cores in your design, such as the GTS System PLL Clocks Intel FPGA IP, GTS Reset Sequencer Intel FPGA IP, TX and RX serial data pins, and the data checker IP.

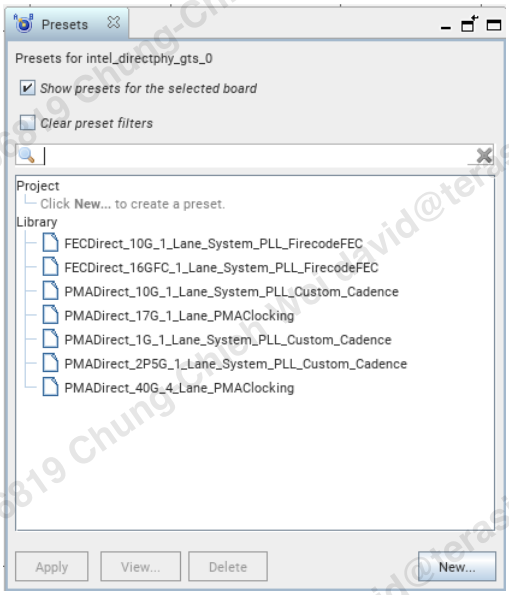
The Intel Agilex 5 E-Series GTS PMA/FEC Direct PHY Intel FPGA IP supports only the following simulators for this release:

- VCS*
- QuestaSim*

3.3.1. Preset IP Parameter Settings

The IP parameter editor provides preset settings for the GTS PMA/FEC Direct PHY Intel FPGA IP. You can specify the preset settings as a starting point for your design. To apply the preset parameters, double-click the preset name, and click **Apply** as shown in the following figure.

Figure 37. Available Parameter Presets In Parameter Editor



For example, selecting the **PMADirect_10G_1_Lane_System_PLL_Custom_cadence** preset enables all parameters and ports that the PMA Direct mode requires, with one GTS PMA operating at 10.3125 Gbps.

Table 16. Preset IP Settings

PMA/FEC Direct Presets	Description
FECDirect_10G_1_Lane_System_PLL_FirecodeFEC	One FEC Direct GTS lane, operating at 10.3125 Gbps with system PLL clocking mode (Firecode FEC enabled)
FECDirect_16GFC_1_Lane_System_PLL_FirecodeFEC	One FEC Direct GTS lane, operating at 14.025 Gbps with system PLL clocking mode (Firecode FEC and custom cadence enabled)
PMADirect_10G_1_Lane_System_PLL_Custom_Cadence	One PMA Direct GTS lane, operating at 10.3125 Gbps with system PLL clocking mode (Custom cadence enabled)
PMADirect_17G_1_Lane_PMAClocking	One PMA Direct GTS lane, operating at 17.16 Gbps with PMA clocking mode

continued...

PMA/FEC Direct Presets	Description
PMADirect_1G_1_Lane_System_PLL_Custom_Cadence	One PMA Direct GTS lane, operating at 1.25 Gbps with system PLL clocking Mode (Custom cadence enabled)
PMADirect_2P5G_1_Lane_System_PLL_Custom_Cadence	One PMA Direct GTS lane, operating at 3.125 Gbps with system PLL clocking Mode (Custom cadence enabled)
PMADirect_40G_4_Lane_PMAClocking	Four PMA Direct GTS lanes, operating at 10.3125 Gbps per lane, with PMA clocking Mode

Specifying a preset removes any existing parameter values for the IP in the parameter editor. Selecting preset parameters does not prevent you from changing any parameter values to meet the requirements of your design.

3.3.2. General and Common Datapath Options

Figure 38. GTS PMA/FEC Direct PHY Intel FPGA IP Parameter Editor

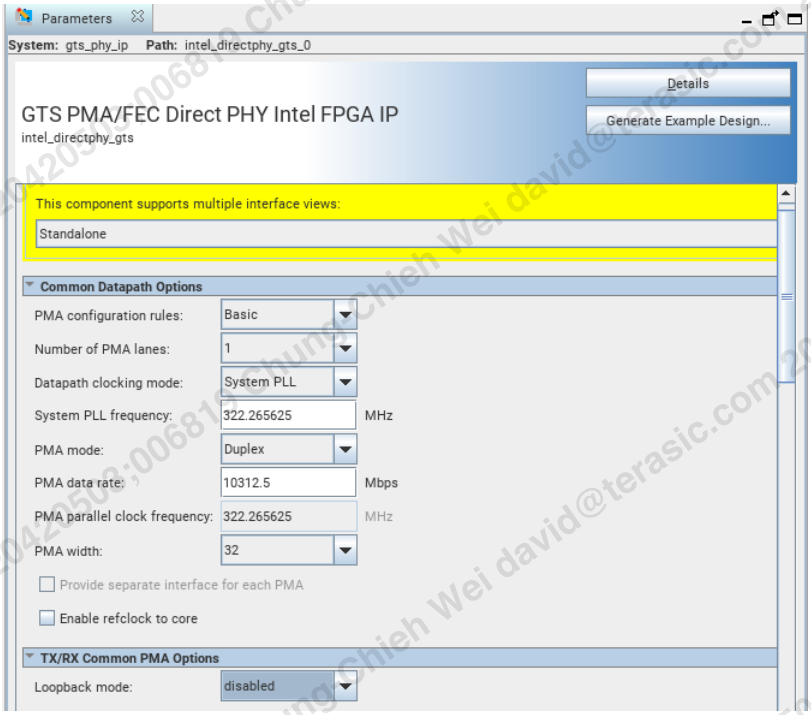


Table 17. Common Datapath Options Parameters

Parameter	Values	Description
PMA configuration rules	**Basic**, **DISPLAY PORT**, **OTN**, **SDI**, **CPRI**, **HDMI**, **CUSTOM**	Selects the protocol configuration rules for the GTS PMA. This parameter governs the rules for correct settings of individual parameters within the PMA. Certain features of the PMA are available only for specific protocol configuration rules. This parameter is not a preset. You must still correctly set all other parameters for your specific protocol and application needs.

continued...

Parameter	Values	Description
		Default value is **Basic**.
Number of PMA Lanes	For TX Simplex and Duplex : **1, 2, 4, 6, 8** For RX Simplex : **1, 2, 3, 4, 6, 8**	Specifies the total number of PMA lanes in a bonded group. For example, if the value is 4, this means there are 4 PMA lanes bonded in the same group and share the same bonding clock. A value of 1 means there is no bonding. The number of PMA lanes is 1 when FEC is enabled. Default value is **1**.
Datapath clocking mode	**PMA** **System PLL**	Specifies whether the PMA parallel clock or System PLL is used to clock the TX/RX datapath. Required to use System PLL when **Enable FEC** is on. Default value is **System PLL**.
System PLL Frequency	**32.5 to 1000 MHz**	Specifies the system PLL clock frequency (MHz) and applicable if datapath clocking mode is selected as system PLL. Default value is **322.265625 MHz**. *Note:* You must ensure that the system PLL frequency and GTS System PLL Clocks Intel FPGA IP frequency is set to the same value if you are using the system PLL clocking mode.
PMA mode	**Duplex, TX Simplex and RX Simplex**	Specifies the PMA operation mode. TX simplex and RX simplex can operate at independent rates at different PMA lanes. Default value is **Duplex**.
PMA Data Rate	E-Series (Device Group B): **10312.5 Mbps** (default) **17160 Mbps** (maximum)	Specifies the PMA data rate in units of Mbps (Mb/sec). Default value for: E-series (Device Group B): **10312.5 Mbps**
PMA Parallel Clock Frequency	**Data rate/PMA width**	Displays PMA parallel clock frequency which is PMA data rate divided by PMA interface width in MHz. Default value is Data rate / PMA Width.
PMA Width	**8, 10, 16, 20, 32**	Specifies the PMA data width. Supported Data width is 8, 10, 16, 20 and 32 bit. Default value is 32
Provide separate interface for each PMA	**On/Off**	When enabled, the GTS PMA/FEC Direct PHY Intel FPGA IP presents separate data and clock interfaces for each PMA lane, rather than a wide bus. Default value is **Off**. *Note:* When the **Enable FEC** option is on, a separate interface is not available for each PMA by use of the **Provide separate interface for each PMA** option.
Enable refclock to core	**On/Off**	Enable the reference clock to FPGA core feature. *Note:* The current release of the Intel Quartus Prime Pro Edition software supports this feature only for one lane.

Table 18. TX/RX Common PMA Options Parameters

Parameter	Values	Description
Loopback mode	**disabled** **parallel**	Selects the PMA loopback mode. Default value is **disabled**.

3.3.3. TX Datapath Options

Figure 39. TX Datapath Options in Parameter Editor

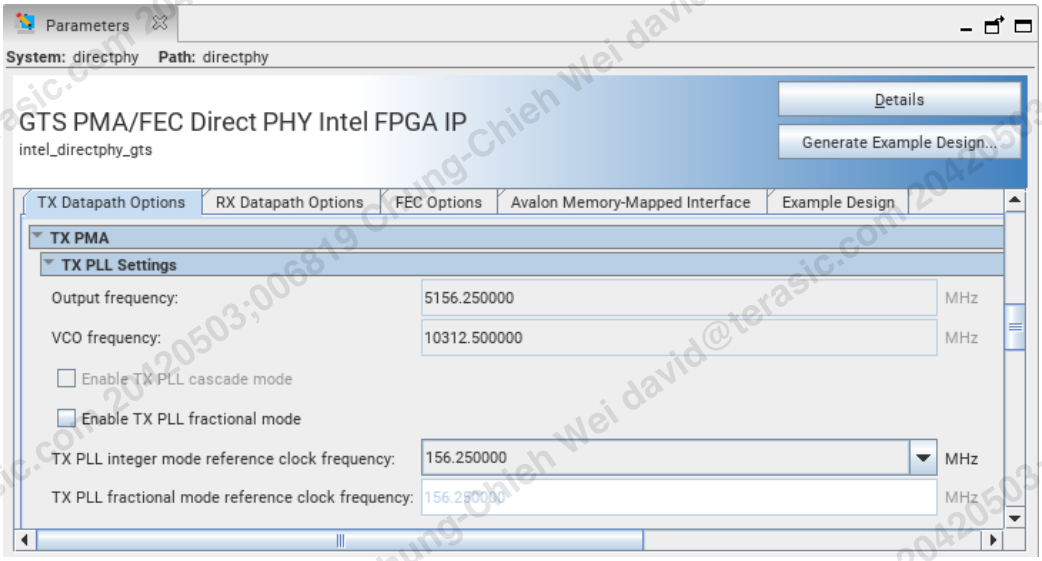


Table 19. TX Datapath Options Parameters

Parameter	Values	Description
TX PMA Parameters		
TX PLL Settings		
Output frequency	N/A	Shows the calculated TX PLL output frequency.
VCO frequency	N/A	Shows the calculated TX PLL VCO output frequency.
Enable TX PLL cascade mode	**On/Off**	Enable cascade mode for Duplex link only. Default value is **Off**. Refer to PMA Support for Fractional Mode for more information.
Enable TX PLL fractional mode	**On/Off**	Enables TX PLL's fractional mode. Default value is **Off**.
TX PLL integer mode reference clock frequency	**40 to 380 MHz**	Selects the reference clock frequency (MHz) for the TX PLL. Range is:

continued...

Parameter	Values	Description
		• 40 – 380 MHz when reference clock is configured for PMA clocking. • 100 – 380 MHz when reference clock is configured for system PLL or shared with system PLL and PMA clocking. • *Note:* Less than 100 MHz is only for HDMI. The reference clock frequency range changes based on the data rate that you select.
TX PLL fractional mode reference clock frequency	**78 to 380 MHz**	Selects the reference clock frequency (MHz) in fractional mode for the TX PLL. Range is: • 78 – 380 MHz when reference clock is configured for PMA clocking. • 100 – 380 MHz when reference clock is configured for system PLL or shared with system PLL and PMA clocking.

3.3.3.1. TX PMA Interface Parameters

Figure 40. TX PMA Interface Parameters in Parameter Editor

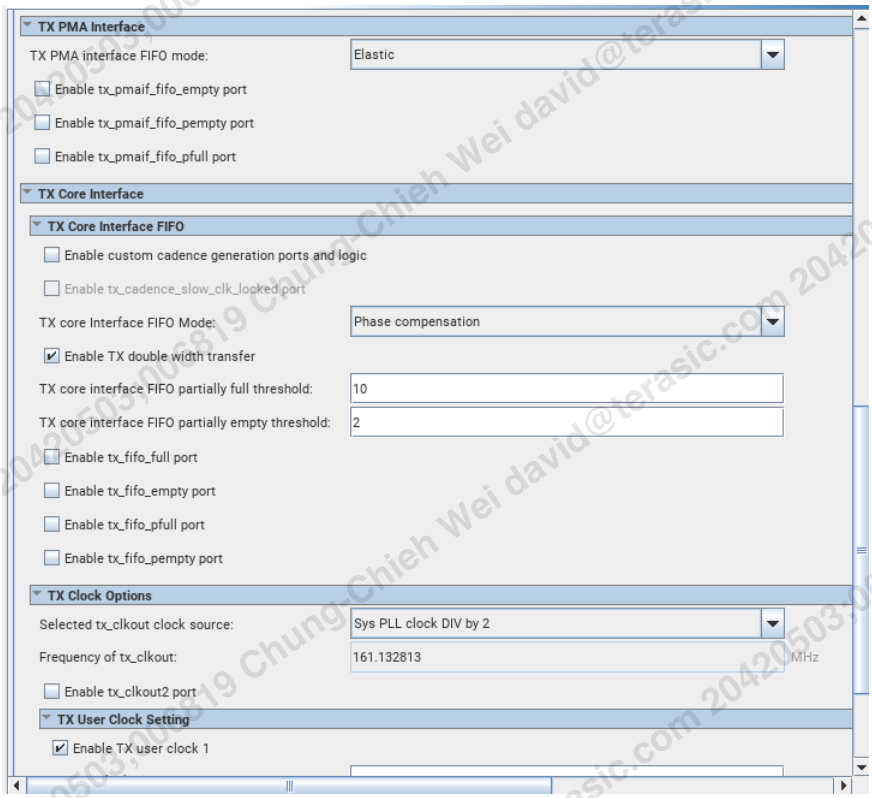


Table 20. TX PMA Interface Parameters

Parameter	Values	Description
TX PMA Interface Parameters		
TX PMA interface FIFO mode	**Phase Compensation** **Elastic**	Selects the TX PMA Interface FIFO mode. Default value is **Elastic**. Refer to PMA Direct Mode Support for more information,
Enable tx_pmaif_fifo_empty port	**On/Off**	Enables the port that indicates the TX PMA Interface FIFO's empty condition. Default value is **Off**.
Enable tx_pmaif_fifo_pfull port	**On/Off**	Enables the port that indicates the TX PMA Interface FIFO's partially full condition. Default value is **Off**.
TX Core Interface Parameters		
Enable custom cadence generation ports and logic	**On/Off**	Enables optional custom cadence generation (CCG) logic and ports (`o_tx_cadence`, `i_tx_cadence_fast_clk`, `i_tx_cadence_slow_clk`). CCG logic can be enabled when **Datapath clocking mode** is set to **System PLL**. Default value is **Off**. Refer to Custom Cadence Generation Ports and Logic for more information.
Enable tx_cadence_slow_clk_locked port	**On/Off**	If `i_tx_cadence_slow_clk` is not directly coming from TX PLL (word clock/TX user clock), but rather comes from another clock source, you must turn on this option in the parameter editor. `i_tx_cadence_slow_clk_locked` port must be driven by the PLL locked output of the other PLL source used for slow clock. Default value is **Off**.
TX core interface FIFO mode	**Phase Compensation** **Elastic**	Specifies the mode for the TX Core Interface FIFO. Default value is **Phase Compensation**. Elastic mode is only supported for PMA Clocking mode.
Enable TX double width transfer	**On/Off**	Enables double width TX data transfer mode. In this mode, the core logic can be clocked with half rate clock. Default value is **Off**.
Enable tx_fifo_full port	**On/Off**	Enables the optional `o_tx_fifo_full` status output port. This signal indicates when the TX core FIFO has reached the full threshold. This signal is synchronous with `o_tx_clkout`. Default value is **Off**.
Enable tx_fifo_empty port	**On/Off**	Enables the optional `o_tx_fifo_empty` status output port. This signal indicates when the TX core FIFO has reached the empty threshold. This signal is synchronous with `o_tx_clkout`. Default value is **Off**.
Enable tx_fifo_pfull port	**On/Off**	Enables the optional `o_tx_fifo_pfull` status output port. This signal indicates when the TX core FIFO has reached the specified partially full threshold. Default value is **Off**.
Enable tx_fifo_pempty port	**On/Off**	Enables the optional `o_tx_fifo_pempty` status output port. This signal indicates when the TX core FIFO has reached the specified partially empty threshold. Default value is **Off**.
TX Clock Options		
Selected tx_clkout clock source	**Word Clock** **TX User Clock** **Sys PLL Clock**	Specifies the `o_tx_clkout` output port source. Default value is **Sys PLL Clock**.

continued...

Parameter	Values	Description
tx_clkout clock div by	**1**, **2**, **4**	Selects the tx clock output divider setting that divides out the `o_tx_clkout` output port source. Default value is **1**.
Frequency of tx_clkout	Output	Displays the frequency of `o_tx_clkout` in MHz based on `o_tx_clkout` source selection.
Enable tx_clkout2 port	**On/Off**	Enables the optional `o_tx_clkout2` output clock. Default value is **Off**.
Selected tx_clkout2 clock source	**Word Clock** **TX User Clock** **Sys PLL Clock**	Specifies the `o_tx_clkout2` output port source. Default value is **Word Clock**.
tx_clkout2 clock div by	**1**, **2**, **4**	Selects the TX clock out 2 divider setting that divides out the `o_tx_clkout2` output port source. Default value is **1**.
Frequency of tx_clkout2	Output	Displays the frequency of `o_tx_clkout2` in MHz based on `o_tx_clkout2` source selection and `o_tx_clkout2` clock divide by factor.
TX User Clock Settings		
TX user clock div by	**12 to 139.5**	Division factor from the Fvco of the TX PLL VCO to TX user clock. Values from 12 to 139.5 are acceptable in 0.5 increments. Default value is **100**.
TX user clock frequency	Output	Displays the frequency of the TX user clock in MHz based on the TX user clock divide by factor.

3.3.4. RX Datapath Options

Figure 41. RX Datapath Options in Parameter Editor

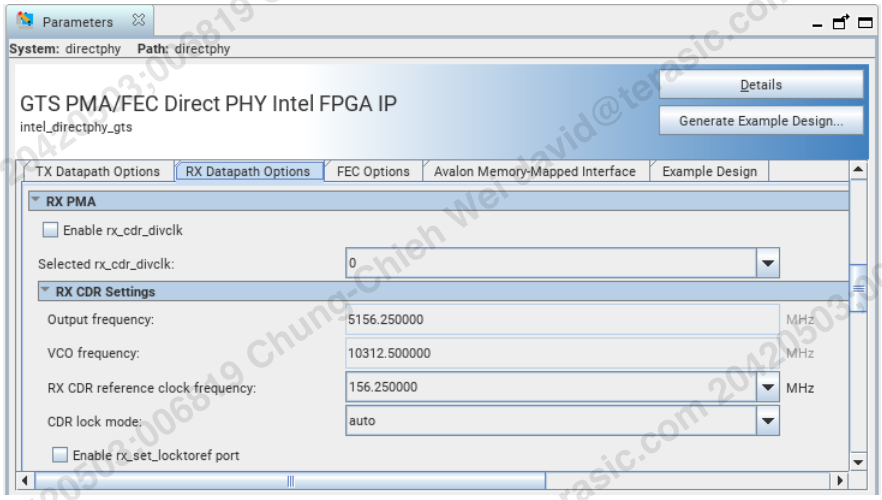


Table 21. RX Datapath Options Parameters

Parameter	Values	Description
Enable rx_cdr_divclk	**On/Off**	Enables the port representing RX CDR clock output from RX PMA to the local reference clock pin (set as output) or the CDR clock output pin. The physical port is typically used for CPRI. Default value is **Off**.
RX CDR Settings		
Output frequency	Output	Specifies the non editable RX CDR output frequency initial value derived from the IP configuration.
VCO frequency	Output	Specifies the non editable RX CDR VCO frequency initial value derived from the IP configuration.
RX CDR reference clock frequency	**25 to 380 MHz**	Selects the reference clock frequency (MHz) for CDR. Default value is **156.25 MHz**.
CDR lock mode	**auto** **manual lock to reference**	When **auto** is selected, during user initiated reset or power-up, CDR first tries to lock to reference and then locks to data if present. By default, loss of lock to data re-triggers reset RX PMA reset. When **manual lock to reference** is selected, you must drive `i_rx_set_locktoref` to control the CDR lock behavior. If `i_rx_set_locktoref` is low CDR operates in **auto** mode, and in lock to reference mode if it is high. When **Enable rx_cdr_divclk** is enabled, only **auto** mode is available. Default value is **auto**.
Enable rx_set_locktoref port	**On/Off**	This parameter is valid only when **CDR lock mode** is set to **manual lock to reference**. Asserting this signal keeps CDR in lock to reference mode. Deasserting this signal keeps CDR in auto mode. When switching modes, assert reset. Default value is **Off**

3.3.4.1. RX PMA Interface Parameters

Figure 42. RX PMA Interface Options in Parameter Editor

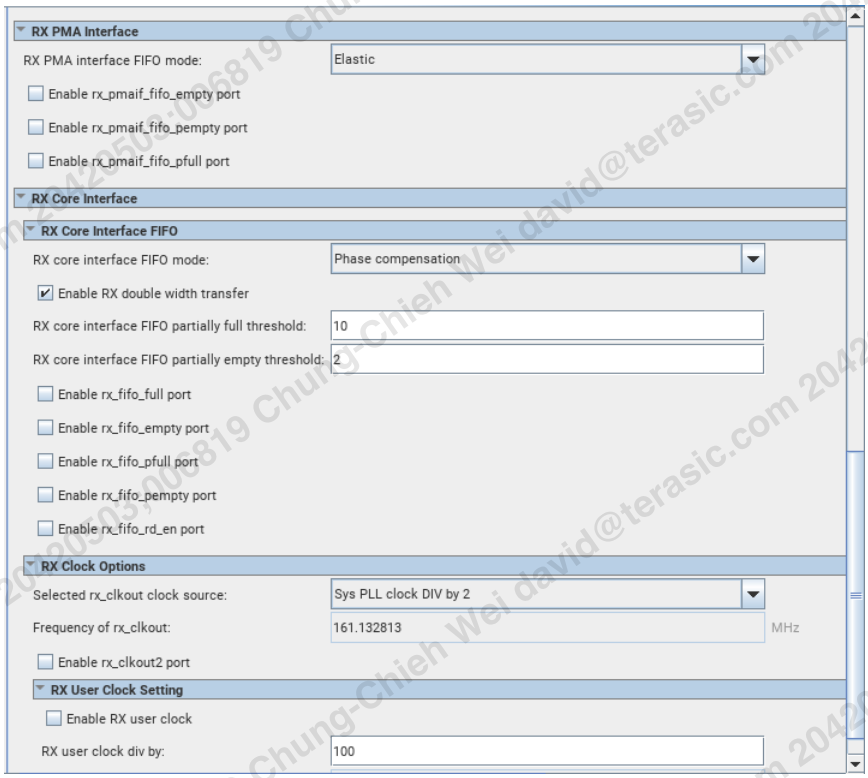


Table 22. RX PMA Interface Parameters

Parameter	Values	Description
RX PMA Interface Parameters		
RX PMA interface FIFO mode	**Register** **Elastic**	Selects the RX PMA Interface FIFO mode. Default value is **Elastic**. Refer to PMA Direct Mode Support for more information,
Enable rx_pmaif_fifo_empty port	**On/Off**	Enables the port that indicates the RX PMA Interface FIFO's empty condition. Default value is **Off**.
Enable rx_pmaif_fifo_pempty port	**On/Off**	Enables the port that indicates the RX PMA Interface FIFO's partially empty condition. Default value is **Off**.
Enable rx_pmaif_fifo_pfull port	**On/Off**	Enables the port that indicates the RX PMA Interface FIFO's partially full condition. Default value is **Off**.
RX Core Interface Parameters		
RX core interface FIFO mode	**Phase compensation** **Elastic**	Specifies the mode for the RX Core Interface FIFO. Default value is **Phase compensation**.
Enable RX double width transfer	**On/Off**	Enables double width RX data transfer mode. In this mode, core logic can be clocked with a half rate clock. Default value is **On**.

continued...

Parameter	Values	Description
Enable rx_fifo_full port	**On/Off**	Enables the optional `o_rx_fifo_full` status output port. This signal indicates when the RX core FIFO has reached the full threshold. This signal is synchronous with `o_rx_clkout`. Default value is **Off**.
Enable rx_fifo_empty port	**On/Off**	Enables the optional `o_rx_fifo_empty` status output port. This signal indicates when the RX core FIFO has reached the empty threshold. This signal is synchronous with `o_rx_clkout`. Default value is **Off**.
Enable rx_fifo_pfull port	**On/Off**	Enables the optionalo_rx_fifo_pfull status output port. This signal indicates when the RX core FIFO has reached the specified partially full threshold. Default value is **Off**.
Enable rx_fifo_pempty port	**On/Off**	Enables the optional `o_rx_fifo_pempty` status output port. This signal indicates when the RX core FIFO has reached the specified partially empty threshold. Default value is **Off**.
Enable rx_fifo_rd_en port	**On/Off**	Enables the optional `i_rx_fifo_rd_en` control input port. This port is used for **Elastic** FIFO mode. Asserting this signal enables the read from RX core FIFO. You must enable this read enable when using Elastic FIFO. Default value is **Off**.
RX Clock Options		
Selected rx_clkout clock source	**Word Clock** **RX User Clock** **Sys PLL Clock**	Specifies the `o_rx_clkout` output port source. Default value is **Sys PLL Clock**.
Frequency of rx_clkout	Output	Displays the frequency of `o_rx_clkout` in MHz based on `o_rx_clkout` source selection.
Selected rx_clkout2 clock source	**Word Clock** **RX User Clock** **Sys PLL Clock**	Specifies the `o_rx_clkout` output port source. Default value is **Word Clock**.
rx_clkout2 clock div by	**1**, **2**, **4**	Selects the RX clock out 2 divider setting that divides out the `o_rx_clkout2` output port source. Default value is **1**.
Frequency of rx_clkout2	Output	Displays the frequency of `o_rx_clkout2` in MHz based on `o_rx_clkout2` source selection and `o_rx_clkout2` clock divide by factor.
RX User Clock Settings		
RX user clock div by	**12-139.5**	Division factor from Fvco of RX CDR to RX user clock. Values from 12 to 139.5 are acceptable in 0.5 increments. Default value is **100**.
RX user clock Frequency	Output	Displays the frequency of RX user clock in MHz based on RX user clock divide by factor.

3.3.5. FEC (Forward Error Correction) Options

The GTS PMA/FEC Direct PHY Intel FPGA IP supports the IEEE 802.3 Firecode (2112, 2080 CL74) FEC mode. You can enable this functionality in the parameter editor by selecting the **Enable FEC** option in the **FEC Options** tab. If **Enable FEC** is off, then **FEC Mode** option is grayed out in the GUI.

Figure 43. FEC Options in Parameter Editor

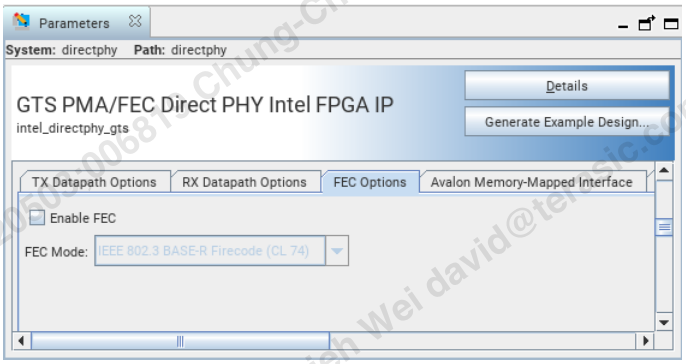


Table 23. FEC Parameters

Parameter	Values	Description
Enable FEC	**On/Off**	Enables or disables the FEC module. Default value is **Off**.
FEC Mode	• IEEE 802.3 Base-R Firecode (CL74)	Specifies the FEC mode for various topologies. Default value is IEEE 802.3 Base-R Firecode (CL74).

3.3.6. Avalon Memory-Mapped Interface Options

Figure 44. Avalon Memory-Mapped Interface Tab in Parameter Editor

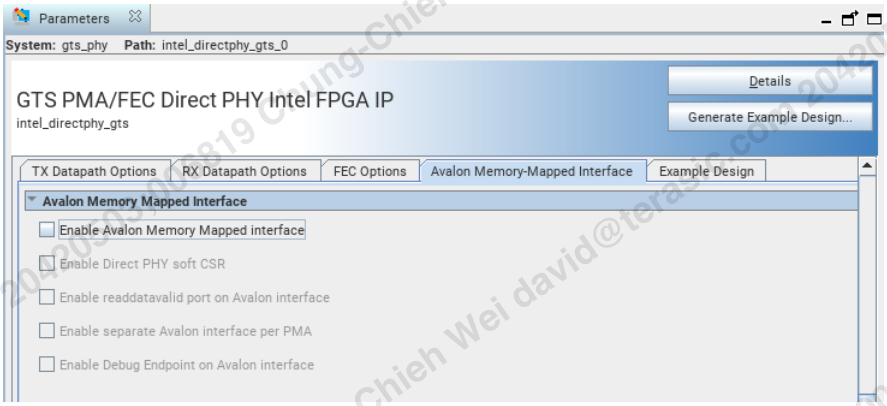


Table 24. Avalon Memory-Mapped Interface Parameters

Parameter	Values	Description
Enable Avalon Memory Mapped Interface	**On/Off**	Enables or disables the Avalon memory mapped interface. Default value is **Off**.
Enable Direct PHY soft CSR	**On/Off**	Enables or disables the soft CSR feature. Default value is **Off**.
Enable readdatavalid port on Avalon interface	**On/Off**	**Off** specifies no `o_reconfig_readdatavalid` port and `o_reconfig_waitrequest` low indicates data valid. **On** specifies `o_reconfig_readdatavalid` port is enabled and indicates data valid. Default value is **Off**.
Enable separate Avalon interface per PMA	**On/Off**	**Off** specifies shared Avalon interface.

continued...

Parameter	Values	Description
		On specifies split interface, if multiple interfaces available with selected targets. Default value is **Off**.
Enable Debug Endpoint on Avalon interface	**On/Off**	When **On**, the GTS PMA/FEC Direct PHY Intel FPGA IP includes an embedded Debug Endpoint that internally connects Avalon memory-mapped agent interface. The Debug Endpoint can access the reconfiguration space of the FEC and the PMA interface block. The IP can perform certain tests and debug functions through JTAG using the System Console. This option may require that you include a `jtag_debug` link in the system. Default value is **Off**.

3.4. Signal and Port Reference

The following section describes all GTS PMA/FEC Direct PHY Intel FPGA IP ports and signals.

Each `tx_parallel_data` and `rx_parallel_data` bus is exposed as 80 bits per lane. Some bits map to special functionality.

Each PMA channel transmits and receives 80 bits, parallel data interface. The determination of active and inactive ports depends on specific configuration parameters, such as the number of lanes and the PMA width.

For details about mapping of data and control signals, refer to *Parallel Data Mapping Information*.

When you enable the **Provide separate interface for each PMA** option for the GTS PMA/FEC Direct PHY Intel FPGA IP, the PHY presents separate data and clock interfaces for each PMA lane, rather than a wide bus. Each PMA lane signal name is appended with a `_xcvr<n>` suffix, with *n = PMA* index number. When **Provide separate interface for each PMA** is disabled, the signal name does not append `_xcvr<n>`.

For example, if you enable **Provide separate interface for each PMA** for two PMA lane configuration, the serial port signal names appear as:

`o_tx_serial_data_xcvr0`, `o_tx_serial_data_xcvr1`.

If you disable **Provide separate interface for each PMA** for two lane PMA configuration, the serial port signal name appears as: `o_tx_serial_data[1:0]`.

The following are the signals that do not have separate interfaces when **Provide separate interface for each PMA** option is on:

- `i_system_pll_clk`
- `i_tx_reset`, `i_rx_reset`, `o_tx_reset_ack`, `o_rx_reset_ack`, `o_tx_ready`, `o_rx_ready`
- `rsfec` signals
- `i_tx_cadence_fast_clk`, `i_tx_cadence_slow_clk`, `i_tx_cadence_slow_clk_locked`

Note: When the **Enable FEC** option is on, a separate interface is not available for each PMA by use of the **Provide separate interface for each PMA** option.

Table 25. Variables Defining Bits for the Interfacing Ports in Port and Signal Reference

Variable	Values	Description
<*N*>	1, 2, 4, 6, 8	*N* is the number of PMA lanes.
<*n*>	0 to *N*-1	*n* is the PMA index number.
<*X*>	*X*=1	*X* is 1 always.
<*M*>	0 to *M*-1	M is the number of banks per side of the device.
<K_p>	Ceiling(log2(N)) K_p = 0, 1, 2, 3, 4 for *N* = 1, 2, 4	K_p is the PMA reconfiguration interface address. K_p=0 if separate Avalon interface per PMA is enabled K_p=Ceiling(log2(*N*) if separate Avalon interface per PMA is disabled.

3.4.1. TX and RX Parallel and Serial Interface Signals

Table 26. TX and RX Parallel and Serial Interface Signals

Refer to Variables Defining Bits for the Interfacing Ports in Port and Signal Reference for variable definitions.

Signal Name	Clocks Domain/ Resets	Direction	Description
i_tx_parallel_data [(80 * *N*)-1:0]	tx_coreclkin tx_reset	input	Parallel data bus from FPGA core to Intel Agilex 5 interface. Some bits map to specific functionality, as *Parallel Data Mapping Information* describes.
o_rx_parallel_data[(80 * *N*)-1:0]	rx_coreclkin rx_reset	output	Parallel data bus from FPGA core to Intel Agilex 5 interface. Some bits map to specific functionality, as *TX and RX Parallel Data Mapping Information for Different Configurations* describes.
o_tx_serial_data[*N*-1:0]	tx_reset	output	TX serial data port. You must assign the port to a TX serial data pin.
o_tx_serial_data_n[*N*-1:0]	tx_reset	output	Differential pair for TX serial data port.
i_rx_serial_data[*N*-1:0]	rx_reset	input	RX serial data port. You must assign the port to a RX serial data pin.
i_rx_serial_data_n[*N*-1:0]	rx_reset	input	Differential pair for RX serial data port.

3.4.2. TX and RX Reference Clock and Clock Output Interface Signals

Table 27. TX and RX Reference Clock and Clock Output Interface Signals

Signal Name	Clocks Domain/ Resets	Direction	Description
o_rx_clkout [(*N*-1):0] o_rx_clkout2 [(*N*-1):0)] o_tx_clkout [(*N*-1):0] o_tx_clkout2 [(*N*-1):0]	N/A	output	Refer to Clock Ports
i_tx_coreclkin [*N*-1:0]	N/A	input	The FPGA core clock. Drives the write side of the TX FIFO.
i_rx_coreclkin [*N*-1:0]	N/A	input	The FPGA core clock. Drives the read side of the RX FIFO.

continued...

Signal Name	Clocks Domain/ Resets	Direction	Description
i_tx_pll_refclk_p[*N*-1:0] i_tx_pll_refclk_n[*N*-1:0]	N/A	input	Reference clock for each of the TX PLL. The local reference clock or the regional reference clock pins must be assigned here. *Note:* Leave the _n signals unconnected and unassigned in your design.
i_rx_cdr_refclk_p[*N*-1:0] i_rx_cdr_refclk_n[*N*-1:0]	N/A	input	Every transceiver bank provides a reference clock input for the RX CDR clock block. The local reference clock or the regional reference clock pins must be assigned here. *Note:* Leave the _n signals unconnected and unassigned in your design.
i_system_pll_clk	N/A	input	To be connected to the GTS System PLL Clock Intel FPGA IP PLL output.
o_tx_pll_locked[*N*-1:0]	asynchronous	output	TX PLL locked signal for reference clock within the PPM threshold status signal. 1'b1 = locked. 1'b0 = not locked.
o_rx_cdr_divclk	N/A	output	This is the clock used for cases like CPRI to bring the recovered clock to an output pin to be used as reference clock.
o_refclk2core[N-1:0]	N/A	output	Transceiver PLL reference clock that you can route to the FPGA fabric, for example for the HDMI use case.
i_system_pll_lock	asynchronous	input	System PLL locked signal

3.4.3. Reset Signals

Table 28. Reset Signals

Signal Name	Clocks Domains	Direction	Description
i_tx_reset	asynchronous	input	TX reset input for TX PMA and TX datapath. Must be kept asserted until o_tx_reset_ack is asserted.
i_rx_reset	asynchronous	input	RX reset input for RX PMA and RX datapath. Must be kept asserted until o_rx_reset_ack is asserted.
o_tx_reset_ack	asynchronous	output	TX fully in reset indicator. This signal asserts following i_tx_reset assertion and stays asserted for as long as i_tx_reset is asserted. This signal deasserts following i_tx_reset deassertion and remains deasserted for as long as i_tx_reset is deasserted.
o_rx_reset_ack	asynchronous	output	RX fully in reset indicator. This signal asserts following i_rx_reset assertion and stays asserted for as long as i_rx_reset is asserted. This signal deasserts following i_rx_reset deassertion and remains deasserted for as long as i_rx_reset is deasserted.

continued...

Signal Name	Clocks Domains	Direction	Description
o_tx_ready	asynchronous	output	Status port to indicate when TX PMA and TX datapath are reset successfully and ready for data transfer.
o_rx_ready	asynchronous	output	Status port to indicate when RX PMA and RX datapath are reset successfully and ready for data transfer.
o_tx_am_gen_start[20]	asynchronous	output	When using FEC, indicates when to start sending alignment markers. This clears after i_tx_am_gen_2x_ack is asserted.
i_tx_am_gen_2x_ack[20]	asynchronous	input	When using FEC, you must indicate to the reset sequencer at least 2 alignment markers were sent since o_tx_am_gen_start is asserted. This signal should be deasserted after o_tx_am_gen_start is deasserted.
o_src_rs_req [N-1:0]	asynchronous	output	Request signal from Soft Reset Controller (SRC) to GTS Reset Sequencer Intel FPGA IP for reset operation. Asserts when there is a request to toggle reset.
i_src_rs_grant [N-1:0]	asynchronous	input	Grant signal from GTS Reset Sequencer Intel FPGA IP to SRC. Asserts when the reset request is granted by Reset Sequencer Intel FPGA IP.
i_pma_cu_clk [M-1:0]	clock	input	PMA Control Unit clock source, one per GTS bank for each side of the device. This clock port must be connected from the GTS Reset Sequencer Intel FPGA IP.

3.4.4. FEC Signals

Table 29. FEC Signals

Signal Name	Clocks Domain/ Resets	Direction	Description
o_fec_status_rx_not_deskew[21]	asynchronous	output	All RX lanes locked but the alignment markers were not unique or the skew was too large. Only applicable in multi-lane.
o_fec_status_rx_not_locked[21]	asynchronous	output	RX lane not locked. Not locked to alignment and codeword markers or RS-FEC codewords (when not using markers). Only applicable in multi-lane.
o_fec_status_rx_not_align[21]	asynchronous	output	Incoming signal fail, RX lanes not all locked, alignment markers not unique or skew too large. Only applicable in multi-lane.
o_fec_sf	asynchronous	output	Signal fail, low means FEC is aligned
i_fec_snapshot	asynchronous	input	Takes a snap of FEC status to CSR, uses Avalon memory-mapped to read the content. Multi-lane FEC is not supported.

[20] This signal is only valid for RS-FEC.

[21] This signal is only valid for RS-FEC.

3.4.5. Custom Cadence Control and Status Signals

Table 30. Custom Cadence Control and Status Signals

Signal Name	Clocks Domain/ Resets	Direction	Description
o_tx_cadence	tx_cadence_fast_clk tx_reset	output	Indicates the rate at which data_valid pin must be asserted and deasserted when the system is running at a higher clock rate than the PMA word clock. Use this signal to assert and de-assert the TX PMA Interface data valid bit when custom cadence generation ports and logic is enabled. Refer to *Parallel Data Mapping Information.*
i_tx_cadence_fast_clk	N/A	input	Fast clock input for tx cadence generator. Use this as the system clock within Intel Agilex 5 device (or use (system clock)/2 when Core Interface is in double width mode). Refer to *Custom Cadence Generation Ports and Logic*..
i_tx_cadence_slow_clk	N/A	input	Slow clock input for tx cadence generator. Use this clock as the PMA word clock (or PMA word clock/2 when Core Interface is in double width mode). Refer to *Custom Cadence Generation Ports and Logic*.
i_tx_cadence_slow_clk_locked	N/A	input	By default, CCG logic assumes i_tx_cadence_slow_clk_locked is coming from TX PLL, and uses o_tx_pll_locked to deassert CGG logic reset. However, if tx_cadence_slow_clk is not directly coming from the TX PLL word clock/user clock), but rather comes from other clock source, then you must turn on the **tx_cadence_slow_clk_locked** port option in the parameter editor. i_tx_cadence_slow_clk_locked must be driven by the PLL locked output of the other clock source used for slow clock.

3.4.6. RX PMA Status Signals

Table 31. RX PMA Status Signals

Signal Name	Clocks Domain/ Resets	Direction	Description
o_rx_is_lockedtoref[*N*-1:0]	asynchronous	output	CDR lock status signal. • 1'b1 – CDR is frequency locked to reference clock within the PPM threshold. • 1'b0 – CDR is not frequency locked within the PPM threshold. Applicable to PMA block only When lockedtodata stays high, the lockedtoref signal status is insignificant.
o_rx_is_lockedtodata[*N*-1:0]	asynchronous	output	RX CDR data lock status signal. • 1'b0: CDR is not locked to data. • 1'b1: CDR is locked to data. Applicable to PMA block.

continued...

Signal Name	Clocks Domain/ Resets	Direction	Description
			When asserted, indicates that the CDR is in locked-to-data mode. When continuously asserted and does not switch between asserted and deasserted, you can confirm that the CDR is actually locked to data.
i_rx_set_locktoref [*N*-1:0]	asynchronous	input	1'b1: keep CDR in lock to reference mode. 1'b0: keep CDR in auto mode.

3.4.7. TX and RX PMA and Core Interface FIFO Signals

Table 32. TX and RX PMA and Core Interface FIFO Signals

Signal Name	Clocks Domain/Resets	Direction	Description
o_tx_pmaif_fifo_empty[(N-1):0]	asynchronous	output	PMA Interface TX FIFO empty.
o_tx_pmaif_fifo_pempty[(N-1):0]	asynchronous	output	PMA Interface TX FIFO partially empty.
o_tx_pmaif_fifo_pfull[(N-1):0]	asynchronous	output	PMA Interface TX FIFO partially full.
o_rx_pmaif_fifo_empty[(N-1):0]	asynchronous	output	PMA Interface RX FIFO empty.
rx_pmaif_fifo_pempty[(N-1):0]	asynchronous	output	PMA Interface RX FIFO partially empty.
o_rx_pmaif_fifo_pfull[(N-1):0]	asynchronous	output	PMA Interface RX FIFO partially full.
o_tx_fifo_full[(N-1):0]	TX Coreclkin TX Reset	output	Core Interface TX FIFO full port.
o_tx_fifo_empty[(N-1):0]	TX Word Clock Sys PLL Clock	output	Core Interface TX FIFO empty port.
o_tx_fifo_pfull[(N-1):0]	TX Coreclkin TX Reset	output	Core Interface TX FIFO partially full port.
o_tx_fifo_pempty[(N-1):0]	TX Word Clock Sys PLL Clock	output	Core Interface TX FIFO partially empty port.
o_rx_fifo_full[(N-1):0]	Transfer clock: Word Clock Sys PLL Clock RX Reset	output	Core Interface RX FIFO full port.
o_rx_fifo_empty[(N-1):0]	RX Coreclkin RX Reset	output	Core Interface RX FIFO empty port.
o_rx_fifo_pfull[(N-1):0]	Transfer clock: Word Clock Sys PLL Clock RX Reset	output	Core Interface RX FIFO partially full port.
o_rx_fifo_pempty[(N-1):0]	RX Coreclkin RX Reset	output	Core Interface RX FIFO partially empty port.
i_rx_fifo_rd_en[(N-1):0]	RX Coreclkin RX Reset	input	Core Interface RX FIFO read enable port.

3.4.8. Avalon Memory Mapped Interface Signals

Table 33. Avalon Memory Mapped Interface Signals (Enable Separate Avalon Interface per PMA = 0)

Signal Name	Clocks Domain/Resets	Direction	Description
i_reconfig_clk	Clock	Input	Reconfiguration Interface Clock. The clock frequency is 100 – 150 MHz.
i_reconfig_reset	i_reconfig_clk	Input	Reconfiguration Interface Reset. Active high reset.
i_reconfig_address[17+K_p:0]	i_reconfig_clk	Input	Reconfiguration Interface Address K_p=Ceiling(log2(N)). Upper address bits are for shared PMA decoding if more than one PMA exists.
i_reconfig_byteenable[3:0]	i_reconfig_clk	Input	Reconfiguration Byte Enable. If byteenable[3:0] is 4'b1111, uses 32-bit Dword Access; otherwise uses byte access.
i_reconfig_write	i_reconfig_clk	Input	Reconfiguration Write
i_reconfig_read	i_reconfig_clk	Input	Reconfiguration Read
i_reconfig_writedata[31:0]	i_reconfig_clk	Input	Reconfiguration Write data
o_reconfig_readdata[31:0]	i_reconfig_clk	Output	Reconfiguration Read data
o_reconfig_waitrequest	i_reconfig_clk	Output	Reconfiguration Wait Request
o_reconfig_readdatavalid	i_reconfig_clk	Output	Reconfiguration Read Data Valid. Optional port, available if the port is enabled in parameter editor.

Table 34. Avalon Memory Mapped Interface Signals (Enable Separate Avalon Interface per PMA = 1)

Signal Name	Clocks Domain/Resets	Direction	Description
i_reconfig_clk<*n*>	Clock	Input	Reconfiguration Interface Clock. The clock frequency is 100 – 150 MHz
i_reconfig_reset<*n*>	i_reconfig_clk<*n*>	Input	Reconfiguration Interface Reset
i_reconfig_address<n>[17:0]	i_reconfig_clk<*n*>	Input	Upper address bits are for shared PMA decoding if more than 1 PMA exists.
i_reconfig_byteenable<*n*>[3:0]	i_reconfig_clk<*n*>	Input	Byte Enable. If byteenable[3:0] is 4'b1111, uses 32-bit Dword; otherwise uses byte access.
i_reconfig_write<*n*>	i_reconfig_clk<*n*>	Input	Reconfiguration Write
i_reconfig_read<*n*>	i_reconfig_clk<*n*>	Input	Reconfiguration Read
i_reconfig_writedata<*n*>[31:0]	i_reconfig_clk<*n*>	Input	Reconfiguration Write data
o_reconfig_readdata<*n*>[31:0]	i_reconfig_clk<*n*>	Output	Reconfiguration Read data
o_reconfig_waitrequest<*n*>	i_reconfig_clk<*n*>	Output	Reconfiguration Wait Request
o_reconfig_readdatavalid<*n*>	i_reconfig_clk<*n*>	Output	Reconfiguration Read Data Valid. Optional port, available if the port is enabled in parameter editor.

3.5. Bit Mapping for PMA and FEC Mode PHY TX and RX Datapath

The `tx_parallel_data` bit and `rx_parallel_data` bit width depends on the **PMA width** and **Number of PMA lanes** IP parameters. Use the following equation to determine the total `tx_parallel_data` or `rx_parallel_data` bit width:

Total tx_parallel_data or rx_parallel_data Bit Width Equation:

```
tx/rx_parallel_data[(80*N)-1:0]
```

Where:

- *N* = Number of PMA lanes value from 1 to 4.

The `tx/rx_parallel_data` signals include the valid parallel data bits and other functionality bits, such as the data valid bit, the write enable for TX core interface FIFO in elastic mode bit, the RX deskew bit, and the alignment marker bits (for FEC mode). These signals travel to and from the FPGA fabric to the transceiver block, and are clocked by the same parallel clock. This parallel clock can be a PMA clock or System PLL clock.

Example 1: Total tx/rx_parallel_data Bit Width with 2 PMA Lanes (*N*=2) and 8-bit PMA Width (*X*=1)

```
tx_parallel_data [(80*2)-1:0] = tx_parallel_data [159:0]
rx_parallel_data [(80*2)-1:0] = rx_parallel_data [159:0]
```

Parallel Data Mapping information for TX and RX

Table 35. Variable Definitions

Variable	Values	Description
N	1, 2, 4, 6, 8	Number of lanes
n	0 to N-1	N is the PMA index number
D	D = PMA Width	D is the data width value to calculate the total parallel data bits

Table 36. PMA Direct Mode Parallel Data Calculations

PMA Configuration	MSB	LSB	TX Parallel Data	RX Parallel Data
PMA Width = 8, 10, 16, 20, 32 **Single Width**	79		Write Enable for TX Core FIFO in Elastic Mode(22)	Data valid for RX Core FIFO in Elastic Mode(22)
	38 + (80*n)		TX PMA Interface Data Valid	RX PMA Interface Data Valid
	[D-1] + (80*n)	0 + (80*n)	TX Data	RX Data
PMA Width = 8, 10, 16, 20, 32 **Double Width**	79		Write Enable for TX Core FIFO in Elastic Mode(22)	Data valid for RX Core FIFO in Elastic Mode(22)

continued...

(22) Only available for PMA clocking and if TX/RX core FIFO is in elastic mode.

PMA Configuration	MSB	LSB	TX Parallel Data	RX Parallel Data
	(40+D-1) + (80*n)	40 + (80*n)	TX Data (Upper Data Bits)	RX Data (Upper Data Bits)
	38 + (80*n)		TX PMA Interface Data Valid	RX PMA Interface Data Valid
	(D -1) + (80*n)	0 + (80*n)	TX Data (Lower Data Bits)	RX Data (Lower Data Bits)

Table 37. FEC Direct Mode Parallel Data Calculations

MSB	LSB	TX Parallel Data	RX Parallel Data
77		Alignment Marker	-
72	40	TX Data (Upper Data Bits)	RX Data (Upper Data Bits)
38		TX PMA Interface Data Valid Bit	RX PMA Interface Data Valid Bit
37		Alignment Marker	Alignment Marker
32	2	TX Data (Lower Data Bits)	RX Data (Lower Data Bits)
1	0	Sync Head	

Table 38. Example of Bit Mapping of TX Parallel Data Bits for PMA Direct Mode with PMA Width = 32

N (Number of Lanes)	1	2	...	4	TX Parallel Data
Bits	79	159		319	Write Enable for TX Core FIFO in Elastic Mode.
	38	118	...	278	TX PMA Interface Data Valid
	31:0	118:80	...	271:240	TX Data (Lower Data Bits)
	71:40	151:120	...	311:280	TX Data (Upper Data Bits)

3.6. Clocking

The GTS transceivers supports three different clock output options from TX and RX that you can use for FPGA core clocking.

Word Clock

The word clock is a PMA parallel clock and equals the data rate divided by the PMA width. For example: 25.78125 Gbps data rate with 32-bit PMA width has a word clock of 25.78125 Gbps ÷ 32 = 805.6640625 MHz.

User Clock

User clock is the divided version of the PMA data rate. The available division factor for user clock is shown below.

The user clock is calculated as the VCO frequency divided by a division factor, which you specify in the **TX/RX user clock div by** parameter in the TX/RX User clock settings in the parameter editor.

```
User clock = VCO frequency / Division factor
```

The valid range of division factors is from 12 to 139.5, in increments of 0.5; for example, 12, 12.5,13,13.5, ……, 139, 139.5.

The TX and RX clocks for word clock and user clock, are two different clocks, derived from TX and RX PMA respectively.

Sys PLL Clock

The Sys PLL clock is the output clock from system PLL. The frequency of this clock is the same as the output frequency of the system PLL connected to the current instance of the GTS PMA/FEC Direct PHY Intel FPGA IP.

3.6.1. Clock Ports

The GTS PMA/FEC Direct PHY Intel FPGA IP supports two clock output ports.

The two clock output ports can each choose one of the three clock options described in *Clock Outputs*.

tx/rx_clkout

`tx/rx_clkout` is an output port that is enabled by default. You can select one of the three clock options described in *Clock Outputs* as the source for this port, by selecting **TX/RX Clock Options ➤ Selected tx/rx_clkout clock source** on the **TX Datapath Options** tab.

tx/rx_clkout2

`tx/rx_clkout2` is an additional output port that you can enable by turning on the **Enable tx/rx_clkout2 port** option in the parameter editor. You can select one of the three clock options as the source for this port, by selecting **TX/RX Clock Options ➤ Selected tx/rx_clkout clock source** on the **TX/RX Datapath Options** tab.

Both the `tx/rx_clkout` and `tx/rx_clkout2` outputs can be further divided by a factor of 1,2, or 4.

Figure 45. tx_clkout and tx_clkout2

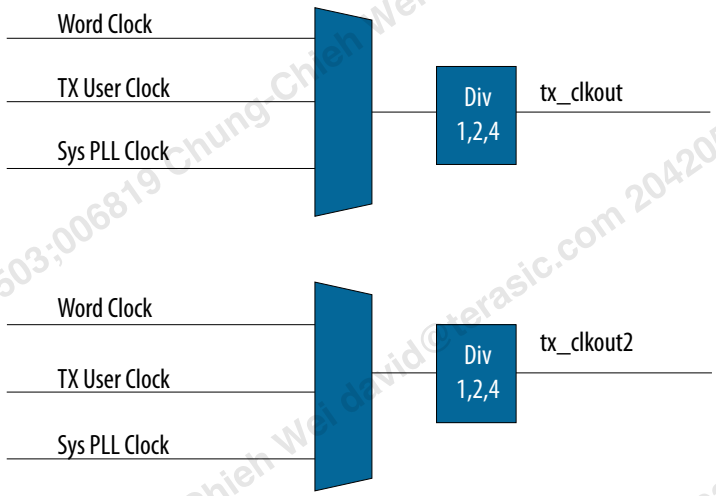
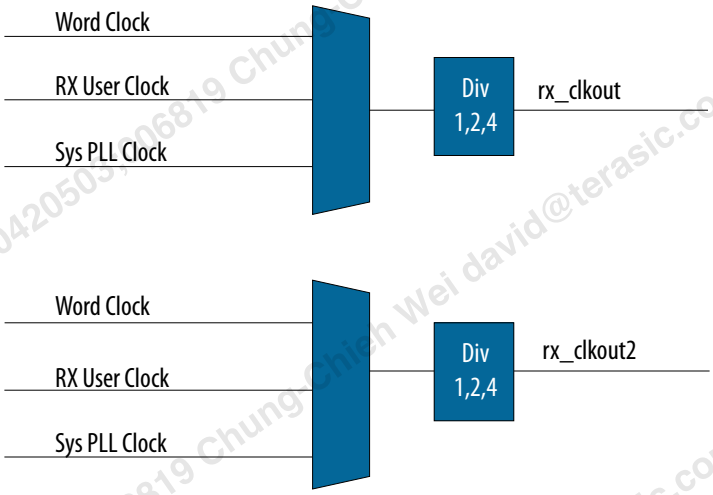


Figure 46. rx_clkout and rx_clkout2



i_tx/rx_coreclkin

`i_tx/rx_coreclkin` is an input port for clocking the TX/RX core interface FIFO. Refer to Recommended Connection and Source for the recommended connections. The recommended source clock for `o_tx/rx_clkout` and `o_tx/rx_clkout2` when connecting to `i_tx/rx_coreclkin` is shown in Recommended tx/rx_coreclkin Connection and tx/rx_clkout2 Source. The recommended port connections details are shown in Port Widths and Recommended Connections for tx/rx_coreclkin, tx/rx_clkout, and tx/rx_clkout2.

3.6.2. Recommended tx/rx_coreclkin Connection and tx/rx_clkout2 Source

Recommended Connection and Source table below shows recommended `i_tx/rx_coreclkin` connection and `o_tx/rx_clkout` and `o_tx/rx_clkout2` source, based on the datapath clocking mode and double-width transfer selection.

Table 39. Recommended Connection and Source

Datapath Clocking Mode	Core Interface FIFO Mode	Enable TX/RX Double Width Transfer	Recommended tx/rx_coreclkin connection	Recommended tx/rx_clkout or tx/rx_clkout2 source	Division factor
PMA	PC	No	`o_tx/rx_clkout`	Word clock	N/A
		Yes	`o_tx/rx_clkout2`	Word clock	2
	Elastic	Yes	`o_tx/rx_clkout2` or any other clock source from user	Word clock/User clock	2

continued...

Datapath Clocking Mode	Core Interface FIFO Mode	Enable TX/RX Double Width Transfer	Recommended tx/rx_coreclkin connection	Recommended tx/rx_clkout or tx/rx_clkout2 source	Division factor
		No	o_tx/rx_clkout or any other clock source from user	Word clock/User clock	N/A
System PLL	PC	No	o_tx/rx_clkout	Sys PLL clock	N/A
		Yes	o_tx/rx_clkout	Sys PLL clock	2

- When using system PLL clocking mode, both `o_tx_clkout` and `o_rx_clkout` can clock `i_tx_coreclkin` and `i_rx_coreclkin`.
- When using PMA clocking mode, `o_tx_clkout/2` must clock `i_tx_coreclkin`. `o_rx_clkout/2` must clock `i_rx_coreclkin`. The only exception to this requirement in PMA clocking mode is for chip to chip applications where TX and RX share same reference clock source (that is, 0 PPM difference), `o_tx_clkout` or `o_rx_clkout` can clock both `i_tx_coreclkin` and `i_rx_coreclkin`.

3.6.3. Port Widths and Recommended Connections for tx/rx_coreclkin, tx/rx_clkout, and tx/rx_clkout2

Port Widths and Recommended Connections shows the port width and recommended connection for the following ports:

- `o_tx_clkout` and `o_rx_clkout`
- `o_tx_clkout2` and `o_rx_clkout2`
- `i_tx_coreclkin` and `i_rx_coreclkin`

Table 40. Port Widths and Recommended Connections

Refer to for variable definitions.

PMA Width	Port Width for tx_clkout, tx_clkout2, tx_coreclkin,rx_clkout, rx_clkout2, rx_coreclkin	Recommended Connection
8, 10, 16, 20, 32	1 * *N*	• Connect `o_tx_clkout[0]` or `o_tx_clkout2[0]` to `i_tx_coreclkin[N-1:0]` • Connect `o_rx_clkout[0]` or `o_rx_clkout2[0]` to `i_rx_coreclkin[N-1:0]`

Table 41. Example with Number of PMA Lane = 1

PMA Width	tx/rx_clkout/2 Port Width	Recommended Connection
8, 10, 16, 20, 32	1	• Connect `o_tx_clkout` or `o_tx_clkout2` to `i_tx_coreclkin` • Connect `o_rx_clkout` or `o_rx_clkout2` to `i_rx_coreclkin`

Table 42. Example with Number of PMA Lanes = 4

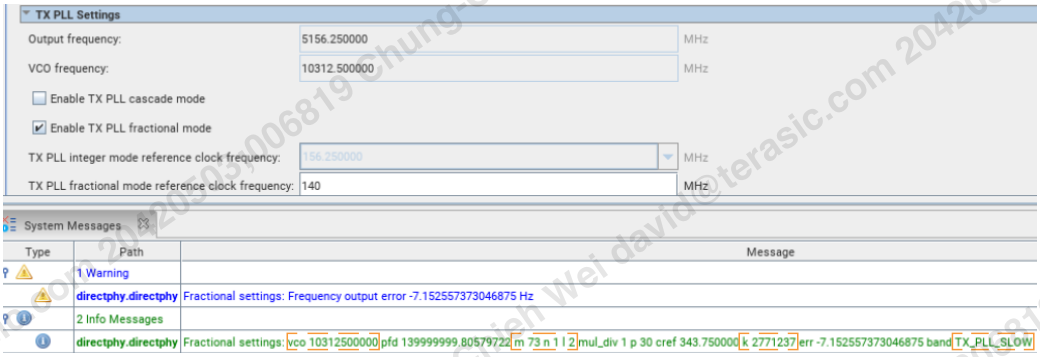
PMA Width	tx/ rx_clkout/2 Port Width	Recommended Connection
8, 10, 16, 20, 32	4	• Connect `o_tx_clkout[0]` or `o_tx_clkout2[0]` to `i_tx_coreclkin[3:0]` • Connect `o_rx_clkout[0]` or `o_rx_clkout2[0]` to `i_rx_coreclkin[3:0]`

3.6.4. PMA Fractional Mode

For a given data rate, the drop-down menu lists the supported integer mode reference clock frequencies. For a given data rate, if the required reference clock frequency is not listed in the drop-down, you can either select one of the supported integer mode reference clock frequencies or enable fractional mode.

- VCO frequency (MHz) = (*M* + *k*/2^22) * *N* * *L* * refclk frequency (MHz). When the fractional PLL reference clock frequency is entered, the IP GUI displays the *VCO, M, N, L, k* values in the **System Messages** tab as shown in the figure below.
- The TX PLL consists of three PLLs: slow, medium and fast. The PLL that is automatically selected is also displayed in the **System Messages** tab as shown in the figure below.
- For a given data rate, you must enable fractional mode if you need to dynamically configure the *K* value during run time. When you enable fractional mode, you must enter the TX PLL fractional mode reference clock frequency.

Figure 47. System Message for PLL Fractional Settings



GTS PMA supports fractional mode in following PMA modes:

Table 43. PMA Support for Fractional Mode

PMA Mode	Fractional Mode Support
TX simplex	Each TX PLL supports fractional mode in TX simplex. To enable, select **TX simplex** option for **PMA mode**, and turn on the **Enable TX PLL fractional mode** in the parameter editor. The TX PLL fractional counter values automatically calculate for the selected reference clock frequency. You can place TX Simplex fractional mode on any of the TX PMAs.

continued...

PMA Mode	Fractional Mode Support
	Note: GTS PMA does not support fractional mode for RX simplex.
Duplex	Each GTS PMA in Duplex PMA mode supports fractional mode. To enable fractional mode in duplex PMA mode, select the **Duplex** option for **PMA mode**, select up to **8** for the **Number of PMA lanes**, and turn on **Enable TX PLL fractional mode** option in the parameter editor. • In **Duplex** fractional mode, the output of each TX PLL is used as the reference clock for corresponding RX CDR. Each TX PLL is configured as fractional mode. • A separate reference clock is not required for RX CDR. TX PLL fractional counter values, and RX CDR reference clock frequency, automatically calculate for the selected reference clock frequency. • You can place duplex fractional mode on any of the PMAs.
Primary PLL configuration	To enable fractional mode with the primary PLL configuration, select the **Duplex** option for **PMA mode**, select **4** for the **Number of PMA lanes**, and turn on **Enable TX PLL fractional mode** and **Enable TX PLL cascade mode** options in the parameter editor. • In Primary PLL configuration, the TX PLL of lane 0 is in fractional mode and acts as the reference clock source for the local CDR and TX PLLs of lanes 1-3 (configured in integer mode) within the transceiver bank. • The TX PLLs in each lane provides the reference clock to the local RX CDR in its own lane. • Primary PLL configuration is only supported when you select 4 for the number of PMA lanes.

Tuning the k Counter Value in Fractional Mode

You can configure the GTS PMA in fractional mode to adjust the frequency and data rate by +/- 1000ppm for rate matching purposes.

Each GTS PMA has an Avalon memory-mapped interface register containing the *K* value. The *K* value / 2^22 gives the fractional value of the feedback counter. The fractional value plus the *M* counter value provides the total feedback counter and determines how much PPM each bit in the *K* value represents.

The procedure to change the *K* value is:

1. Change the *K* value to the new value.
2. Pulse the strobe bit 0 -> 1-> 0 to lock in the new *K* value.

Each GTS PMA contains 3 PLLs; slow, medium and fast. The *K* value and strobe bit Avalon memory-mapped interface register addresses depend on the location of the transceiver in the bank and which PLL is used (slow, medium, fast) as shown in the table below.

Table 44. GTS PMA Fractional *k* Counter and Strobe Register Addresses

Channel Location in Bank	PLL	Fractional *K* Value Register	Strobe Register
0	Slow	0x094000[30:9]	0x09400C[17]
	Medium	0x094100[30:9]	0x09410C[17]
	Fast	0x094200[30:9]	0x09420C[17]
1	Slow	0x194000[30:9]	0x19400C[17]
	Medium	0x194100[30:9]	0x19410C[17]
	Fast	0x194200[30:9]	0x19420C[17]
2	Slow	0x294000[30:9]	0x29400C[17]

continued...

Channel Location in Bank	PLL	Fractional *K* Value Register	Strobe Register
	Medium	0x294100[30:9]	0x29410C[17]
	Fast	0x294200[30:9]	0x29420C[17]
3	Slow	0x394000[30:9]	0x39400C[17]
	Medium	0x394100[30:9]	0x39410C[17]
	Fast	0x394200[30:9]	0x39420C[17]

3.6.5. Core PLL Mode

You can configure the TX PLL in core PLL mode to provide a pixel clock for the Display Port protocol. You cannot use the PMA (both TX and RX) for normal operation, when you use the TX PLL in core PLL mode.

To use the core PLL mode:

- Select **DISPLAY PORT** in the **PMA configuration rules** drop down list under the **Common Datapath Options** in the GTS PMA/FEC Direct PHY Intel FPGA IP parameters.
- Set the **Datapath clocking mode** to **PMA**.
- Select the TX simplex PMA mode.
- Specify the PMA data rate of 10,000 Mbps.
- Select the TX PLL reference clock frequency of 150 MHz.
- Specify TX user clock divide by value.

You must select `tx_clkout` clock source as user clock. The calculated output frequency is shown in the frequency of the `tx_clkout` parameter.

The output frequency of the TX PLL is the VCO frequency divided by TX user clock divide by value. You can modify the TX user clock divide by value to get the required output frequency.

The Core PLL mode which routes the clock to the FPGA core fabric can only be used for Display Port pixel clock generation, and for no other purpose. This mode is only supported when the **DISPLAY PORT** is selected under **PMA configuration rules**.

3.7. Custom Cadence Generation Ports and Logic

When using system PLL clocking mode, you must enable the **Custom cadence generation (CCG) ports and logic** parameter for the use cases that the *Custom Cadence Generation Ports and Logic Use Cases* table below describes. Enabling CCG logic ensures that the TX PMA interface FIFO does not overflow due to the over clocking of the datapath when using system PLL clocking mode.

Table 45. Custom Cadence Generation Ports and Logic Use Cases

Configuration	Datapath Clocking mode	System PLL Frequency	Enable Custom Cadence Generation (CCG) Ports and Logic
PMA Direct	PMA	N/A	No
PMA Direct	System PLL	Equal to PMA parallel clock frequency. No PPM between PMA parallel clock frequency and system PLL frequency. That is, the same reference clock source for PMA and system PLL.(23)	No
PMA Direct	System PLL	Greater than the PMA parallel clock frequency.	Yes
FEC Direct	System PLL	Equal to the PMA Parallel clock frequency. No PPM between PMA parallel clock frequency and system PLL frequency. That is, the same reference clock source for PMA and system PLL.	No
FEC Direct	System PLL	Equal to the PMA Parallel clock frequency. PPM between PMA parallel clock frequency and system PLL frequency. That is, different reference clock for PMA and system PLL.	Yes
FEC Direct	System PLL	Greater than the PMA parallel clock frequency.	Yes

When you enable **Custom cadence generation (CCG) ports and logic**, the `o_tx_cadence`, `i_tx_cadence_fast_clk`, and `i_tx_cadence_slow_clk` ports are available in the GTS PMA/FEC Direct PHY Intel FPGA IP. CCG logic uses the `i_tx_cadence_fast_clk` and `i_tx_cadence_slow_clk` inputs (does not monitor PMA Interface FIFO status), and generates a `o_tx_cadence` output signal. You must use `o_tx_cadence` to assert and de-assert the TX PMA Interface data valid bit. This bit is one of the bits in TX parallel data. Refer to *Parallel Data Mapping Information*.

Table 46. tx_cadence_fast_clk and tx_cadence_slow_clk connections

Configuration	Enable TX Double Width Transfer	Recommended Connections
PMA Direct	Yes	• Connect `i_tx_cadence_fast_clk` to System PLL Clock Div2 • Connect `i_tx_cadence_slow_clk` to word clock / 2
PMA Direct	No	• Connect `i_tx_cadence_fast_clk` to System PLL Clock • `i_tx_cadence_slow_clk` to word clock
FEC Direct	Yes	• Connect `i_tx_cadence_fast_clk` to System PLL Clock Div2 • `i_tx_cadence_slow_clk` to User Clock (DIV 66 or DIV 68)

3.7.1. Enabling the `i_tx_cadence_slow_clk_locked` Port

If the `i_tx_cadence_slow_clk` signal does not come directly from TX PLL (word clock or user clock), but rather comes from the other clock source (as might be applicable in FEC Direct modes when using slower clock to accommodate FEC

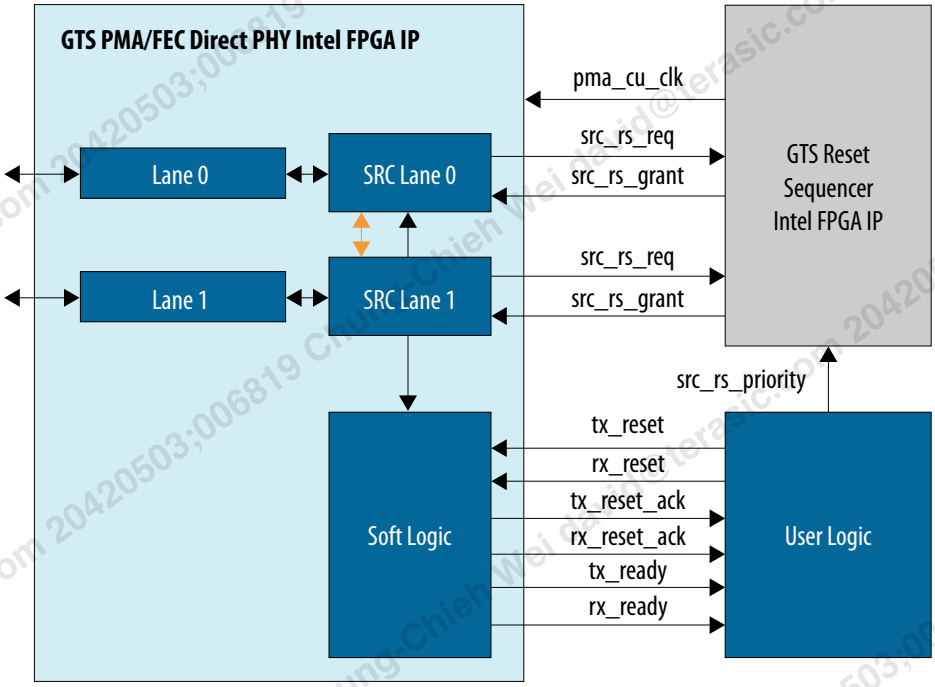
(23) When using PMA direct with system PLL clocking mode, if the reference clock for PMA and system PLL are from different clock source, then the system PLL frequency cannot be equal to the PMA parallel clock frequency. System PLL frequency must be greater than or equal to the fastest possible TX and RX PMA clock, including PPM.

overhead), you must enable the `tx_cadence_slow_clk_locked` port in the IP parameter editor. The PLL locked output of the other clock source used for slow clock must drive `i_tx_cadence_slow_clk_locked`.

3.8. Asserting reset

The Soft Reset Controller (SRC) together with GTS Reset Sequencer Intel FPGA IP handles all non-PCIe reset scheduling and sequencing for the Intel Agilex 5 FPGAs in the PMA Direct and FEC Direct modes. The SRC is automatically instantiated inside the GTS PMA/FEC Direct PHY Intel FPGA IP based on the channels that are used whereas GTS Reset Sequencer Intel FPGA IP is a mandatory IP that you must manually instantiate for your design. Refer to Implementing the GTS Reset Sequencer Intel FPGA IP for more information. Asserting a reset sequence ensures that the PMA in each channel initializes and functions correctly. You can reset the transmitter (TX) and receiver (RX) data paths independently or together.

Figure 48. Reset Top-Level Diagram



3.8.1. Reset Signal Requirements

The following requirements apply to reset signal use for the GTS PMA/FEC Direct PHY FPGA IP designs:

- Ensure that `i_tx_reset`/`i_rx_reset` remain asserted until `o_tx_reset_ack`/`o_rx_reset_ack` goes+ high.
- Expect random data if `o_tx_ready`/`o_rx_ready` is not asserted.
- In forward error correction (FEC) modes, during reset sequencing, after `o_tx_am_gen_start` is asserted, start sending alignment markers and assert `i_tx_am_gen_2x_ack` after two alignment markers are sent. `o_tx_am_gen_start` goes high as part of reset sequence (that is, before `o_tx_ready` is asserted).
- In FEC modes when sending alignment markers, you can pace tx data valid with the `o_tx_cadence` signal.
- For duplex configurations, you can assert `i_tx_reset` and `i_rx_reset` independently.

3.8.2. Power On Reset Requirements

Power On reset requirements are the same as for run-time reset.

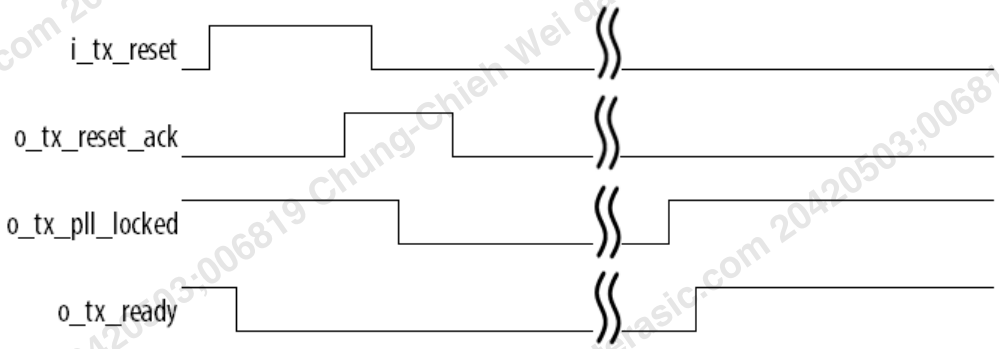
3.8.3. Reset Signals—Block Level

Table 47. Reset Signals—Block Level

Reset Signal	TX PMA	TX Datapath	RX PMA	RX Datapath	Soft CSRs	PMA Reconfiguration Interface
`i_tx_reset`	Yes	Yes	No	No	No	No
`i_rx_reset`	No	No	Yes	Yes	No	No
`i_reconfig_reset`	No	No	No	No	Yes	Yes

3.8.4. Run-time Reset Sequence—TX

Figure 49. Run-time Reset Sequence—TX

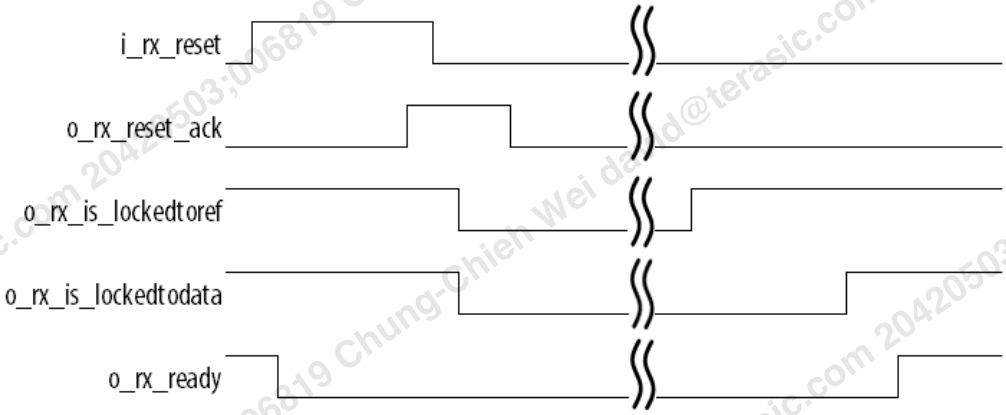


The figure above illustrates the following run-time TX reset sequence:[24]

1. Assert `i_tx_reset`.
2. `o_tx_ready` deasserts, indicating that the TX datapath is no longer operational.
3. `o_tx_reset_ack` asserts, indicating that the TX datapath is fully in reset. `o_tx_reset_ack` stays asserted until `i_tx_reset` deasserts.
4. `o_tx_pll_locked` deasserts.
5. You then deassert `i_tx_reset` to bring TX out of reset.
6. `o_tx_pll_locked` asserts as the TX PLL locks to the reference clock.
7. `o_tx_ready` asserts.

3.8.5. Run-time Reset Sequence—RX

Figure 50. Run-time Reset Sequence—RX



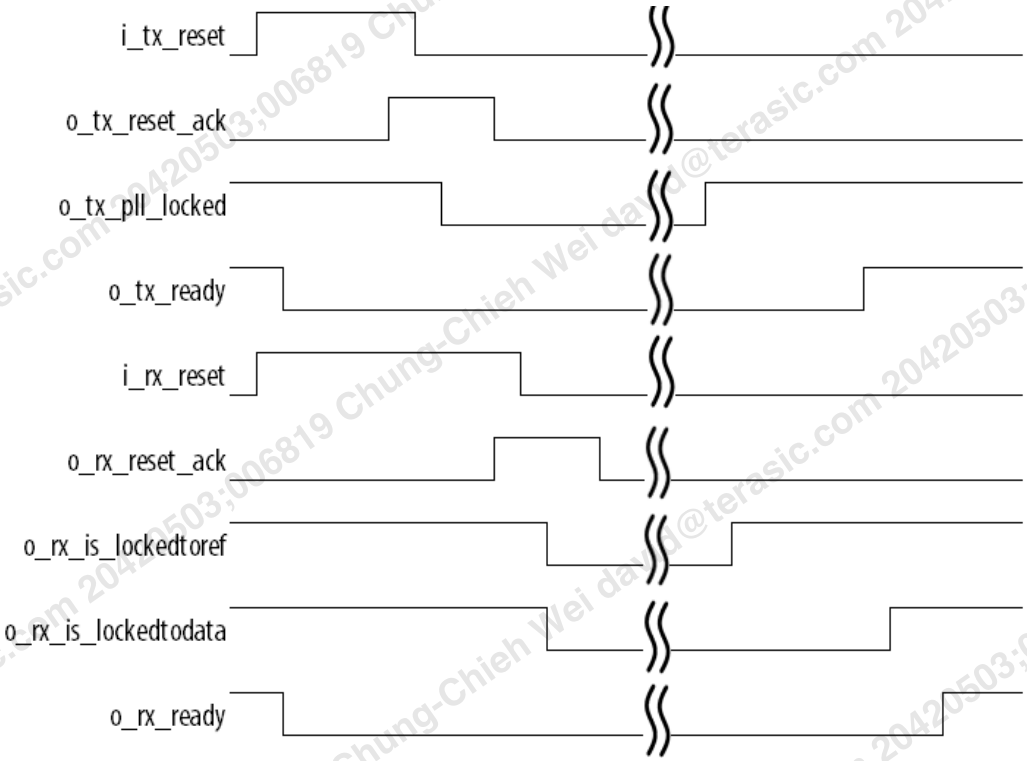
The figure above illustrates the following run-time RX reset sequence:

1. Assert `i_rx_reset`.
2. `o_rx_ready` deasserts, indicating that the RX datapath is no longer operational.
3. `o_rx_reset_ack` asserts, indicating that the RX datapath is fully in reset. `o_rx_reset_ack` stays asserted until `i_rx_reset` deasserts.
4. `o_rx_is_lockedtoref` and `o_rx_is_lockedtodata` deasserts.
5. You then deassert `i_rx_reset` to bring RX out of reset.
6. `o_rx_is_lockedtoref` asserts as the CDR locks to the reference clock.
7. `o_rx_is_lockedtodata` asserts as the CDR locks to the recovered data.
8. `o_rx_ready` asserts.

(24) All timing diagrams show relative signal behavior and the waves do not show actual durations in time.

3.8.6. Run-time Reset Sequence—TX + RX

Figure 51. Run-time Reset Sequence—TX + RX

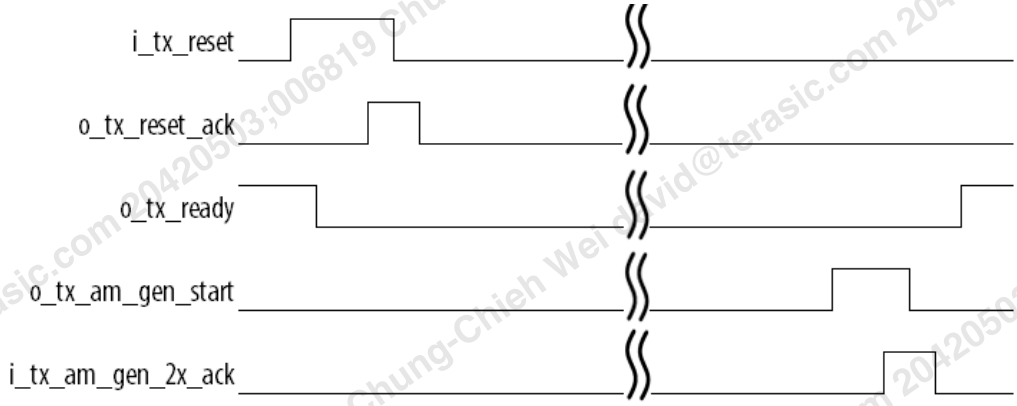


The figure above illustrates the following run-time TX - RX reset sequence:

1. Assert `i_tx_reset` and `i_rx_reset`.
2. `o_tx_ready` and `o_rx_ready` deassert, indicating that the datapaths are no longer operational.
3. `o_tx_reset_ack` and `o_rx_reset_ack` assert, indicating that the datapaths are in reset. `o_tx_reset_ack` and `o_rx_reset_ack` stay asserted until i_tx_reset and i_rx_reset deassert.
4. `o_tx_pll_locked` and `o_rx_is_lockedtodata` deassert.
5. You then deassert `i_tx_reset` and `i_rx_reset`.
6. `o_tx_pll_locked` asserts as the PLL locks to the reference clock.
7. `o_rx_is_lockedtoref` asserts as the CDR locks to the reference clock.
8. `o_rx_is_lockedtodata` asserts as the CDR locks to the recovered data.
9. `o_tx_ready` and `o_rx_ready` assert, indicating that the TX and RX datapaths are ready for use.

3.8.7. Run-time Reset Sequence—TX with FEC

Figure 52. Run-time Reset Sequence—TX with FEC



As illustrated in the above figure, the following is the run-time reset sequence for TX with forward error correction (FEC):

1. Assert `i_tx_reset`.
2. `o_tx_ready` deasserts, indicating that the TX datapath is no longer operational.
3. `o_tx_reset_ack` asserts, indicating that the TX datapath is in reset.
4. You then deassert `i_tx_reset` to bring TX out of reset.
5. `o_tx_am_gen_start` asserts, you then send at least two alignment markers on the `tx_parallel_data` bus.
6. Assert `i_tx_am_gen_2x_ack`, indicating that at least two alignment markers have been sent.
7. `o_tx_am_gen_start` deasserts, you then deassert `i_tx_am_gen_2x_ack`.
8. `o_tx_ready` asserts.

3.9. Bonding Implementation

Bonding enables you to minimize skew between channels. The bonding implementation in Intel Agilex 5 FPGA supports x2, x4, x6, and x8 channels of the PMA.

Bonding is automatically enabled when more than one PMA lane is enabled in the GTS PMA/FEC Direct PHY Intel FPGA IP. The Intel Quartus Prime software implements bonding automatically. Following table shows the parameter setting to achieve bonding or no bonding.

Table 48. Bonding Implementation

Bonding Implementation	Number of PMA Lanes
No Bonding	= 1
Bonding	> 1

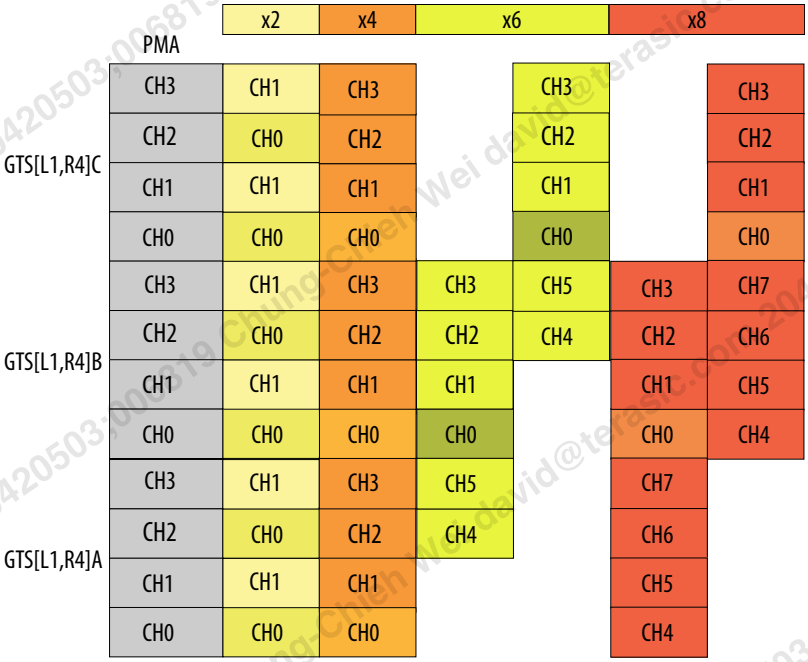
When bonding is enabled and the Datapath clocking mode is set to **PMA**, select **Word Clock** for the **Selected tx_clkout clock source** parameter in the **TX Clock Options** group box and also for the **Selected rx_clkout clock source** in the **RX Clock Options** group box.

When bonding is enabled and the Datapath clocking mode is set to **System PLL**, select either **Sys PLL Clock** or **Sys PLL Clock** with division by 2 for the **Selected tx_clkout clock source** and **Selected rx_clkout clock source** parameters, respectively.

3.9.1. Bonding Placement Requirements

For the Intel Agilex 5 bonding implementation, you must follow the channel placement guidelines shown in the following figure for the channels in the bonded groups (x2, x4, x6, and x8). In the figure, CH0-CH7 are the logical channel number of the various bonded groups.

Figure 53. Bonding Channel Placement Requirement Guidelines



3.10. Configuration Register

Each GTS transceiver channel has an Avalon memory-mapped interface for reconfiguration. You can access the GTS PMA/FEC Direct PHY Intel FPGA IP soft CSRs and the GTS PMA registers using the same Avalon memory-mapped interface.

Here are the key considerations for the configuration registers:

- Write operations to a read-only register field have no effect.
- Write operations to reserved registers have an undefined effect.
- Read operations that address a reserved register return an unspecified result.
- Accesses to registers that do not exist in your IP core variation, or to register bits that your IP core variation does not define, have an unspecified result. You must consider these registers and register bits reserved.
- Although you can only access registers in 32-bit read and write operations, do not attempt to write or ascribe meaning to values in undefined register bits.

The GTS PMA register map contains the reconfiguration register information for:

- PMA and FEC Direct PHY soft CSR registers
- GTS PMA registers

The following sections describe the register map for each area and how to access the registers.

3.10.1. GTS PMA and FEC Direct PHY Soft CSR Register Map

The *GTS PMA and FEC Direct PHY Soft CSR Register Map* allows you to read out the status of the GTS PMA/FEC Direct PHY Intel FPGA IP configuration settings, Avalon memory-mapped ready signals, PMA ready signals, TX PLL locked and RX CDR lock-to-data status signals. It also allows you to control settings for the PMA hard and soft reset signals.

In order to access the soft CSR registers, you must enable following options in the **Avalon Memory-Mapped Interface** tab of the GTS PMA/FEC Direct PHY Intel FPGA IP parameter editor:

- **Enable Avalon Memory Mapped interface**
- **Enable Direct PHY soft CSR**

Note: You can select the **Enable Debug Endpoint on Avalon Interface** parameter, if you plan to use the GTS PMA/FEC Direct PHY Intel FPGA IP debug interconnect fabric to connect the Direct PHY soft CSR registers with the JTAG interface. Refer to Using Debug Endpoint Interface within the GTS PMA/FEC Direct PHY Intel FPGA IP for more information on accessing this Avalon interface.

The starting address for the GTS PMA/FEC Direct PHY Intel FPGA IP soft CSR register through the Avalon memory-mapped interface is 0x800h. You can refer to the *PMA and FEC Direct PHY Soft CSR Register* tab in the *GTS PMA/FEC Direct PHY Intel FPGA IP Register Map* for more details.

Note: The GTS PMA/FEC Direct PHY Intel FPGA IP only has one Avalon memory-mapped interface that can access the entire address space.

Related Information

GTS PMA/FEC Direct PHY Intel IP Register Map

3.10.2. GTS PMA Register Map

The *GTS PMA Register Map* consists of the PMA Analog registers, TX PLL counter registers, debug and loopback register information for the GTS PMA lanes.

In order to access GTS PMA registers, you must enable following option in the **Avalon Memory-Mapped Interface** tab of the GTS PMA/FEC Direct PHY Intel FPGA IP parameter editor; **Enable Avalon Memory Mapped interface**.

Note: You can select the **Enable Debug Endpoint on Avalon Interface** parameter, if you plan to use the GTS PMA/FEC Direct PHY Intel FPGA IP debug interconnect fabric to connect the GTS PMA registers with the JTAG interface. Refer to Using Debug Endpoint Interface within the GTS PMA/FEC Direct PHY Intel FPGA IP for more information on accessing this Avalon interface.

Related Information

GTS PMA/FEC Direct PHY Intel IP Register Map

3.10.3. Logical Avalon Memory-Mapped Port Indexing

This section explains how to access the GTS PMA lanes if your design has more than one GTS PMA lane. You need to know how to control the logical Avalon memory-mapped port indexing if you are using the shared reconfiguration interface. (**Enable separate Avalon interface per PMA = Off**).

If your design has the **Enable separate Avalon interface per PMA = On** feature enabled, each Avalon memory-mapped interface has its own reconfiguration interface and you can directly access the port's register address without following the logical port indexing instruction explained in this section.

The reconfiguration address bus for one PMA lane in the IP is:

- 18 bits for the PMA Avalon memory-mapped reconfiguration interface (`i_reconfig_address[17:0]`)

The reconfiguration address space grows as a function of the following equation: $\log_2(N)$

(where N = number of lanes), for an increase in the number of PMA lanes. The additional MSB bits of the reconfiguration address bus represent the logical Avalon memory-mapped port index value. Refer to MSB Address Bits for PMA Logical Avalon Memory-Mapped Reconfiguration Port Index Value for more information.

For example, if you enable four PMA lanes in your design, for the Avalon memory-mapped interface, the total reconfiguration address bus is `i_reconfig_address[19:0]`. The Avalon memory-mapped interfaces MSB address bits (`i_reconfig_address[19:18]`) provide the logical mapping to access different lanes based on the number of PMA lanes. The three additional MSB address bits (`i_reconfig__address[19:18]`) are a logical representation of the Avalon memory-mapped interface port. You can use them to read or write to the individual Avalon memory-mapped interface defined within the IP.

The following table shows the MSB address bits for logical Avalon memory-mapped port indexing with the number of PMA lanes configured in the IP.

Table 49. MSB Address Bits for Logical Avalon Memory-Mapped Reconfiguration Port Index Value

PMA Lane Set in GUI(s)	Reconfiguration Address Bus	MSB Address Bits for Logical Avalon Memory-Mapped Port Indexing	Logical Avalon Memory-Mapped Port Indexing (value=hex)
1	i_reconfig_address[17:0]	Not Applicable	0
2	i_reconfig_address[18:0]	[18]	0,1
4	i_reconfig_address[19:0]	[19:18]	0,1,2,3
6, 8	i_reconfig_address[20:0]	[20:18]	0,1,2,3,4,5,6,7

3.10.4. Accessing Configuration Registers

This section summarizes how to access the configuration registers listed in the *GTS PMA/FEC Direct PHY Intel FPGA IP* register map. You can use the detailed information to access the PMA and FEC Direct PHY Soft CSR registers and GTS PMA registers. In this section, the offset address terminology refers to the address of the configuration registers in the *GTS PMA/FEC Direct PHY Intel FPGA IP* register map.

3.10.4.1. Accessing GTS PMA and FEC Direct PHY Soft CSR Registers

For the GTS PMA and FEC Direct PHY Soft CSR registers, you can directly use the offset address shown in the *GTS PMA/FEC Direct PHY Intel FPGA IP* register map. For example, to monitor the TX PLL lock status, you must use address `0x810`.

3.10.4.2. Accessing GTS PMA Registers

The following table shows the offset address between lanes that you must add when you want to access the registers for a design with more than one lane. Note that the word address is byte address/4.

Table 50. Lane Number and Address Offset

Lane Number	Offset (Byte Address)
0	0x000000
1	0x100000
2	0x200000
3	0x300000

For example, if you want to update the TX equalizer co-efficients settings for the GTS PMA lanes within a bank, refer to the SRDS_IP_IF_TX1 register (`0x097830`) in the register map file and add `0x100000h` for each incremental lane, as shown below:

- For Lane 0: `0x097830`
- For Lane 1: `0x197830`
- For Lane 2 : `0x297830`
- For Lane 3 : `0x397830`

Note: You can access each GTS PMA channel's registers in a bank through the same base address. For the example shown, all lanes use the same base address of 0x097830.

3.10.4.3. Accessing PMA Lane Number Information

The following table shows the offset address between lanes that you must add when you want to access the registers for a design with more than one lane. Note that the word address is byte address/4.

Table 51. Lane Number and Address Offset

Lane Number	Offset (Byte Address
0	0x000000
1	0x100000
2	0x200000
3	0x300000

For example, if you want to read the lane number information for the GTS PMA lanes within a bank, refer to the GTS_LANE_Number register (0x0A5000) in the register map file and add 0x100000h for each incremental lane, as shown below:

- For Lane 0: 0x0A5000
- For Lane 1: 0x1A5000
- For Lane 2 : 0x2A5000
- For Lane 3 : 0x3A5000

Note: Lane 0, 1, 2, or 3 are the physical locations where the channels are placed and corresponds to CH0, CH1, CH2, and CH3 respectively. You can add an incremental offset of 0x100000 to this address to access up to lane 7 (0x7A5000) to read the physical GTS PMA lane information (if you enable 8 GTS PMA lanes in your design per side and do not **Enable Separate Avalon interface per PMA** feature in the GTS PMA/FEC Direct PHY Intel FPGA IP).

3.11. Configuring the GTS PMA/FEC Direct PHY Intel FPGA IP for Hardware Testing

This section details the steps you should follow to configure the GTS PMA/FEC Direct PHY Intel FPGA IP in order to bring-up the GTS PMA for hardware testing using System Console in the Intel Quartus Prime software. You can use these methods to configure the GTS PMA analog settings to enable functions such as serial loopback, PRBS generators and checkers, to modify TX equalizer settings, and BER measurements.

You can choose either of the following methods to access the PMA registers via JTAG using System Console:

- Using Debug Endpoint Interface within the GTS PMA/FEC Direct PHY Intel FPGA IP
- Using JTAG to Avalon Master Bridge Intel FPGA IP

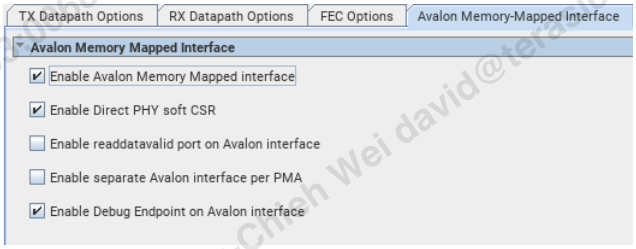
3.11.1. Using Debug Endpoint Interface within the GTS PMA/FEC Direct PHY Intel FPGA IP

The Debug Endpoint Avalon interface is a JTAG Avalon memory-mapped interface that provides access to the reconfiguration register space of the GTS PMA through System Console. The Intel Quartus Prime software inserts the debug interconnect fabric to connect the GTS PMA with JTAG.

To enable the Debug Endpoint Avalon Interface, follow these steps:

1. In the **Avalon Memory-Mapped Interface** tab of the GTS PMA/FEC Direct PHY Intel FPGA IP parameter editor, enable the following options:
   - **Enable Avalon Memory Mapped interface**
   - **Enable Direct PHY soft CSR**
   - **Enable Debug Endpoint on Avalon interface**

Figure 54. Avalon Memory-Mapped Interface Parameter Settings to Enable Debug Endpoint



2. Connect the clock and reset signals to the `i_reconfig_clk` and `i_reconfig_reset` ports of the reconfiguration interface.
3. Connect the other reconfiguration interface signals:
   - `i_reconfig_write`
   - `i_reconfig_read`
   - `i_reconfig_address`
   - `i_reconfig_writedata`
   - `i_reconfig_byteenable`

   to ground, assuming no FPGA core logic controls the reconfiguration interface.

   Note: If you do not connect the reconfiguration interface signals appropriately, the debug endpoint functions unexpectedly.

3.11.1.1. RTL Connection Example for Debug Endpoint Avalon Interface

The following example shows the RTL connections for a single GTS PMA channel with clock and reset connections and no FPGA core logic driving the additional reconfiguration ports.

Figure 55. Example RTL Connections for a One GTS PMA Lane Design

```
.i_reconfig_clk          ( i_reconfig_clk    ),  // 100 MHz
.i_reconfig_reset        ( i_reconfig_reset  ),
.i_reconfig_write        ( 1'b0              ),
.i_reconfig_read         ( 1'b0              ),
.i_reconfig_address      ( 17'b0             ),
.i_reconfig_byteenable   ( 4'b0              ),
.i_reconfig_writedata    ( 32'b0             ),
.o_reconfig_readdata     (                   ),
.o_reconfig_waitrequest  (                   )
```

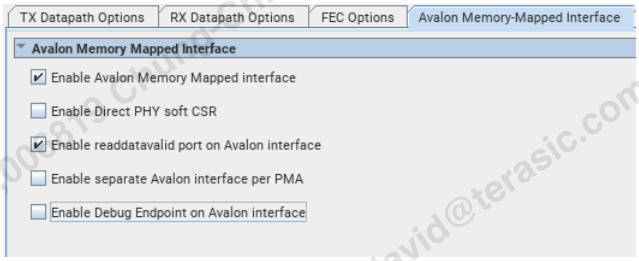
3.11.2. Using JTAG to Avalon Master Bridge Intel FPGA IP

The JTAG to Avalon Master Bridge Intel FPGA IP provides access to the reconfiguration register space of the GTS through System Console. The Intel Quartus Prime software inserts the debug interconnect fabric to connect the JTAG interface to the GTS PMA.

To Enable the JTAG to Avalon Master Bridge Intel FPGA IP Interface, follow these steps:

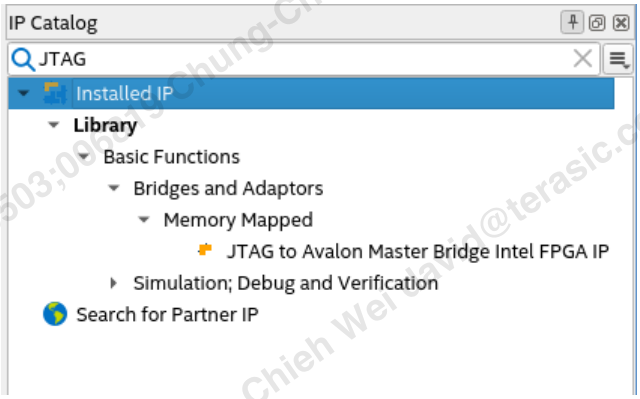
1. In the **Avalon Memory-Mapped Interface** tab of the GTS PMA/FEC Direct PHY Intel FPGA IP parameter editor, enable the following options:
   - **Enable Avalon Memory Mapped interface**
   - **Enable readdatavalid on Avalon interface**

Figure 56. Avalon Memory-Mapped Interface Parameter Settings to Enable JTAG to Avalon Master Bridge



2. Instantiate JTAG to Avalon Master Bridge Intel FPGA IP instance from the **IP Catalog**. You use the IP to interface with the Avalon interface.

Figure 57. IP Catalog



3. Connect the clock and reset signals to the `i_reconfig_clk` and `i_reconfig_reset` ports of the reconfiguration interface.
4. Connect the other reconfiguration interface signals:
   - `i_reconfig_write`
   - `i_reconfig_read`
   - `i_reconfig_address`
   - `i_reconfig_writedata`
   - `o_reconfig_readdata`
   - `i_reconfig_byteenable`
   - `o_reconfig_readdatavalid`
   - `o_reconfig_waitrequest`

   to the equivalent JTAG to Avalon Master Bridge Intel FPGA IP reconfiguration signals.

3.11.2.1. RTL Connection Example for JTAG to Avalon Master Bridge Intel FPGA IP

The following example shows the RTL connections for a two lane GTS channel implementation.

The `i_reconfig_address` address bus width is:

```
[18:0]i_reconfig_address
```

You must set the address parameters for the interface in your design file as follows:

```
parameter  addr_width = 19
```

Note: The GTS PMA/FEC Direct PHY Intel FPGA IP uses word addressing format for the reconfiguration address bus interface. The JTAG to Avalon Master Bridge Intel FPGA IP uses byte addressing format. Therefore, you must handle the conversion of word addressing format to byte addressing format by shifting the reconfiguration address (`i_reconfig_address`) bus by two bits, as shown in the following example.

Figure 58. Example RTL Connections for the JTAG to Avalon Master Bridge Intel FPGA IP

```
.i_reconfig_clk           ( i_reconfig_clk        ),  // 100 MHz
.i_reconfig_reset         ( i_reconfig_reset      ),
.i_reconfig_write         ( write_bridge          ),
.i_reconfig_read          ( read_bridge           ),
.i_reconfig_address       ( address_bridge [addr_width + 1: 2]),
.i_reconfig_byteenable    ( byteenable_bridge     ),
.i_reconfig_writedata     ( writedata_bridge      ),
.o_reconfig_readdata      ( readdata_bridge       ),
.o_reconfig_readdatavalid ( readdatavalid_bridge  ),
.o_reconfig_waitrequest   ( waitrequest_bridge    )
```

3.12. Configurable Intel Quartus Prime Software Settings

You can configure the GTS PMAs using the Intel Quartus Prime software settings file. (`.qsf`)

You can specify values for the following HSSI parameters in the Intel Quartus Prime settings file (`.qsf`) or use the **Assignment Editor** of the Intel Quartus Prime Pro Edition software to configure the GTS PMAs:

TX Equalization:

```
set_instance_assignment -name HSSI_PARAMETER "tx_eq_main_tap=<parameter_value>" -
to <TX_SERIAL_PIN> -entity <TOP_LEVEL_NAME>
```

Valid parameter settings:

- main_tap: 0-55
- pre_tap_1: 0-15
- pre_tap_2: 0-7
- post_tap: 0-19

Table 52. TX Equalization HSSI Parameter Name and Values

HSSI Parameter Name	Valid Parameter Values (Decimal)
tx_eq_main_tap	0-55
tx_eq_pre_tap_1	0-15
tx_eq_pre_tap_2	0-7
tx_eq_post_tap_1	0-19

Example assignments in `.qsf` file:

- ```
  set_instance_assignment -name HSSI_PARAMETER "tx_eq_main_tap=41" -to
  c12tx_serial[0] -entity top
  ```
- ```
  set_instance_assignment -name HSSI_PARAMETER "tx_eq_pre_tap_1=1" -to
  c12tx_serial[0] -entity top
  ```
- ```
  set_instance_assignment -name HSSI_PARAMETER "tx_eq_pre_tap_2=0" -to
  c12tx_serial[0] -entity top
  ```
- ```
  set_instance_assignment -name HSSI_PARAMETER "tx_eq_post_tap_1=4" -to
  c12tx_serial[0] -entity top
  ```

Figure 59. Assignment Editor Example for TX Equalization

To	Assignment Name	Value	Enabled	Entity
out c12tx_serial[0]	Transceiver Parameter	tx_eq_main_tap=26	Yes	top
out c12tx_serial[0]	Transceiver Parameter	tx_eq_post_tap_1=0	Yes	top
out c12tx_serial[0]	Transceiver Parameter	tx_eq_pre_tap_1=6	Yes	top
out c12tx_serial[0]	Transceiver Parameter	tx_eq_pre_tap_2=0	Yes	top

Note: It is recommended that you set some baseline values for the TX equalization parameters for your design to optimize the GTS PMA transmitter.

RX AC coupling:

```
set_instance_assignment -name HSSI_PARAMETER
"rx_external_couple_type=<parameter_value>" -to <RX_SERIAL_PIN> -entity
<TOP_LEVEL_NAME>
```

Valid parameter settings:

- RX_EXTERNAL_COUPLE_TYPE_AC: When you enable external AC coupling capacitors in your link.
- RX_EXTERNAL_COUPLE_TYPE_DC: When you do not enable external AC coupling capacitors in your link.

Table 53. RX AC Coupling HSSI Parameter Name and Values

HSSI Parameter Name	Valid Parameter Values	Use Case
`rx_external_couple_type`	RX_EXTERNAL_COUPLE_TYPE_AC	When AC coupling capacitor is enabled externally in the link.
`rx_external_couple_type`	RX_EXTERNAL_COUPLE_TYPE_DC	When AC coupling capacitor is not enabled externally in the link.

Example assignment in `.qsf` file:

```
set_instance_assignment -name HSSI_PARAMETER
"rx_external_couple_type=RX_EXTERNAL_COUPLE_TYPE_AC" -to c12rx_serial[0] -entity
top
```

Figure 60. Assignment Editor Example for RX AC Coupling

To	Assignment Name	Value	Enabled	Entity
in c12rx_serial[0]	Transceiver Parameter	rx_external_couple_type=RX_EXTERNAL_COUPLE_TYPE_AC	Yes	top

Note: It is recommended that you set the `rx_external_couple_type` parameter based on your transmission link coupling type.

RX on-chip termination:

```
set_instance_assignment -name HSSI_PARAMETER
"rx_onchip_termination_setting=<parameter_value>" -to <RX_SERIAL_PIN> -entity
<TOP_LEVEL_NAME>
```

Valid parameter settings:

- RX_ONCHIP_TERMINATION_SETTING_R_1: 85 Ohms
- RX_ONCHIP_TERMINATION_SETTING_R_2: 100 Ohms

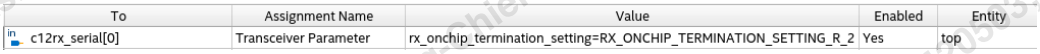
Table 54. RX On-Chip Termination HSSI Parameter Name and Values

HSSI Parameter Name	Valid Parameter Values	Use Case
rx_onchip_termination_setting	RX_ONCHIP_TERMINATION_SETTING_R_1	85 Ohm
rx_onchip_termination_setting	RX_ONCHIP_TERMINATION_SETTING_R_2	100 Ohm

Example assignment in `.qsf` file:

```
set_instance_assignment -name HSSI_PARAMETER
"rx_onchip_termination_setting=RX_ONCHIP_TERMINATION_SETTING_R_2" -to
c12rx_serial[0] -entity top
```

Figure 61. Assignment Editor Example for RX On-Chip Termination



Note: It is recommended that you set the `rx_onchip_termination_setting` parameter based on your transmission link characteristics.

4. Implementing the GTS System PLL Clocks Intel FPGA IP

The GTS System PLL Clocks Intel FPGA IP is a required IP for the GTS PMA/FEC Direct PHY Intel FPGA IP or any other protocol IPs that use system PLL clocking.

GTS System PLL Clocks Intel FPGA IP Overview

The GTS System PLL Clocks Intel FPGA IP performs the function described below:

- Configures the system PLL:
  - Enable system PLL and specify the mode
  - Specifies the output and reference clock of the system PLL

This IP does not configure the IOPLL that can be used as a second system PLL in devices with a single transceiver bank. Refer to Intel Agilex 5 Clocking and PLL User Guide for more information.

4.1. IP Parameters

The table below lists the IP parameters for the GTS System PLL Clocks Intel FPGA IP.

Figure 62. GTS System PLL Clock Intel FPGA IP Parameter Editor

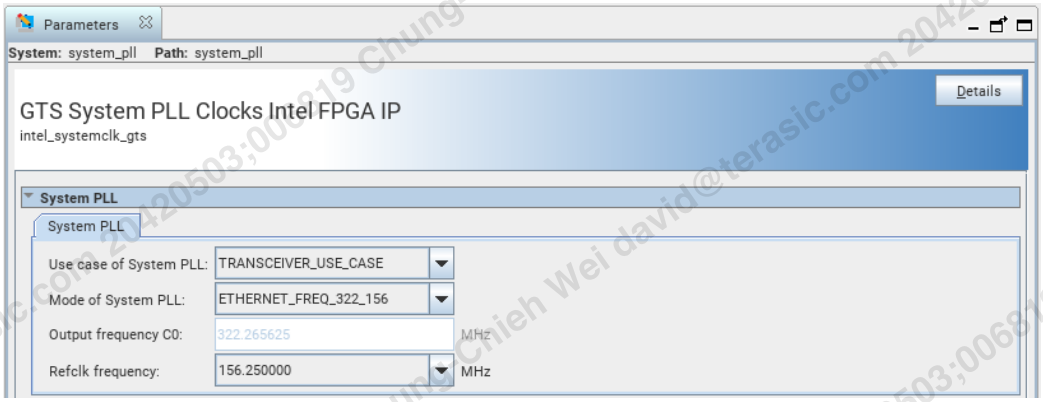


Table 55. GTS System PLL Clock Intel FPGA IP Parameters

Parameter	Values	Description
System PLL		
Use case of system PLL	**TRANSCEIVER_USE_CASE**	Use case of system PLL. use the **TRANSCEIVER_USE_CASE** to supply clock to the transceivers.
Mode of system PLL	**User Configuration**	Selects the mode of system PLL.

continued...

Parameter	Values	Description
	User PCIe-based Configuration **ETHERNET_FREQ_322_156** **ETHERNET_FREQ_322_322** **PCIE_FREQ_250** **PCIE_FREQ_275** **PCIE_FREQ_300** **PCIE_FREQ_325** **PCIE_FREQ_350** **PCIE_FREQ_375** **PCIE_FREQ_400** **PCIE_FREQ_425** **PCIE_FREQ_450** **PCIE_FREQ_475**	• **User configuration**— manually configure the output frequency of the system PLL and input reference clock frequency. For use in non-PCIe use cases when other Ethernet presets do not meet your requirements. • **User PCIe-based configuration** — manually configure the output frequency of system PLL and input reference clock frequency. For use in PCIe use cases when the PCIe presets do not meet your requirements. • **ETHERNET_FREQ_**<*output-freq*>_<*refclk-freq*> — presets for Ethernet use cases. *output_freq* is the system PLL output frequency and *refclk_freq* is the system PLL reference clock frequency. • **PCIE_FREQ_**<*output-freq*> — presets for PCIe use cases. *output_freq* is the system PLL output frequency. *Note:* The frequency number in the preset labels are abbreviated; they are not the full precise frequencies. Refer to the *Preset Reference Clock and Output Frequencies* table for the full frequencies. The default value is **ETHERNET_FREQ_322_156**.
Refclk frequency	**25.78125 MHz to 380 MHz**	Specifies the reference clock frequency.
Output frequency C0	**31.25 MHz to 1000 MHz**	Specifies the output frequency of the **system PLL C0** in MHz. In background, the algorithm calculates the legal reference clock frequencies for that clock output frequency. For correct calculation, specify the exact frequency with decimal points. *Note:* You must ensure that the output frequency of the system PLL and the GTS PMA/FEC Direct PHY Intel FPGA IP are set to the same frequency if you are using the system PLL clocking mode.

Related Information

- GTS Ethernet Intel FPGA Hard IP User Guide
- GTS PCI Express Intel FPGA IP

4.2. IP Port List

The following table lists the ports for the IP; all ports are 1-bit wide.

Table 56. GTS System PLL Clock Intel FPGA IP Port List

Port Name	Direction	Description
i_refclk	Input	Reference clock input port. Must be assigned to device reference clock pin. This port can be connected to the local or regional reference clock pins, or the reference clock pins from the HVIO bank, described in System PLL with HVIO Reference Clock. Refer to the device pinout

continued...

Port Name	Direction	Description
		documentation for the available pins. If this port connects to differential pins, you must connect the positive signal of the differential pair to this input port.
i_refclk_ready	Input	Reference clock ready indicator port. This port is available only when user selects a non-PCIe mode. Refer to the PCIe IP user guide for more information on the PCIe mode.
o_pll_lock	Output	System PLL lock output port. System PLL lock status port which indicates if system PLL is locked to incoming reference clock.
o_syspll_c0	Output	System PLL clock output c0 port. This must be connected to system PLL clock input of protocol IP.

4.3. Mode of System PLL - System PLL Reference Clock and Output Frequencies

Table 57. Preset Reference Clock and Output Frequencies

Mode of System PLL - System PLL	Reference Clock (MHz)	Output Frequency (MHz)
PCIE_FREQ_250	100	250
PCIE_FREQ_275	100	275
PCIE_FREQ_300	100	300
PCIE_FREQ_325	100	325
PCIE_FREQ_350	100	350
PCIE_FREQ_375	100	375
PCIE_FREQ_400	100	400
PCIE_FREQ_425	100	425
PCIE_FREQ_450	100	450
PCIE_FREQ_475	100	475
ETHERNET_FREQ_322_156	156.25	322.265625
ETHERNET_FREQ_322_322	322.265625	322.265625

Table 58. Port Connection Guidelines between GTS System PLL Clock Intel FPGA IP and GTS PMA/FEC Direct PHY Intel FPGA IP

GTS System PLL Clock Intel FPGA IP	GTS PMA/FEC Direct PHY Intel FPGA IP
System PLL	
o_syspll_c0	i_system_pll_clk
o_pll_lock	i_system_pll_lock

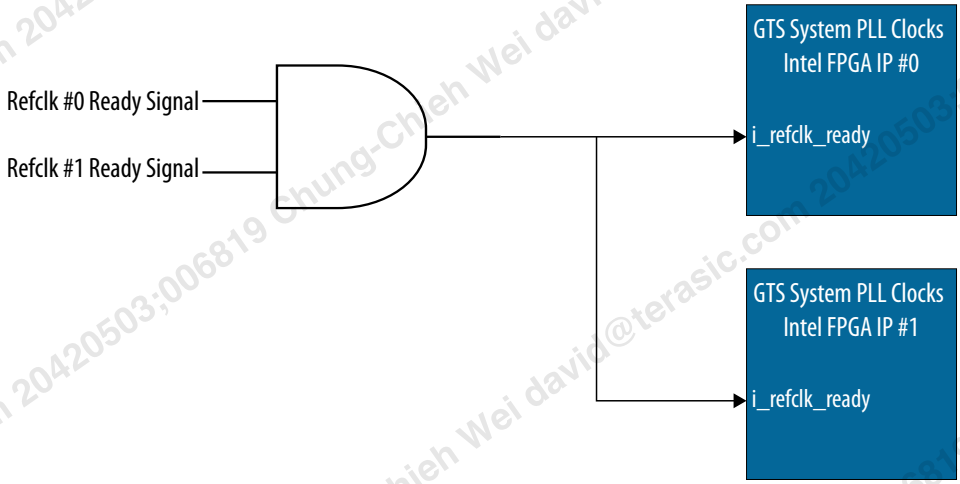
4.4. Guidelines for GTS System PLL Clocks Intel FPGA IP Usage

You must adhere to the following guidelines to correctly use the GTS System PLL Clock Intel FPGA IP:

- The GTS System PLL Clock Intel FPGA IP cannot be compiled or simulated as a standalone IP. When you use the GTS System PLL Clock Intel FPGA IP, it must always connect to the GTS PMA/FEC Direct PHY Intel FPGA IP or protocol IPs.
- You must connect the system PLL output ports of GTS System PLL Clocks Intel FPGA IP to input of GTS PMA/FEC Direct PHY Intel FPGA IP as shown in *Port Connection Guidelines between GTS System PLL Clock Intel FPGA IP and GTS PMA/FEC Direct PHY Intel FPGA IP* or protocol IPs.
- You must ensure the reference clock and system PLL frequencies specified in GTS System PLL Clocks Intel FPGA IP match reference clock and system PLL frequencies specified in GTS PMA/FEC Direct PHY Intel FPGA IP or protocol IPs.
- You must instantiate one GTS System PLL Clocks Intel FPGA IP for every system PLL you intend to use in the design.

- Each system PLL can be used by the channels in its own transceiver bank, or by channels in the transceiver banks immediately above or below its own transceiver bank. The location of the system PLL is automatically assigned by Intel Quartus Prime Pro Edition software.
- You must inform the IP when all reference clocks are ready after device configuration is complete.
  - An input port `i_refclk_ready` is available, and this port must be set high once the reference clock is ready after device configuration. If this port is not set high, the system PLL does not attempt to lock to the reference clock, and the `o_pll_lock` status output does not go high.
  - You can connect this input port to a GPIO pin to control this externally. You can also control this input port internally by setting it from your RTL logic.
  - If the reference clock signal is ready before device configuration, this input port can be tied high.
  - You must bring up all the reference clocks in your design that feed the system PLLs before any of the GTS transceivers are used. You can do a logical and of all the reference clock ready signals for multiple GTS System PLL Clocks Intel FPGA IPs together as shown in the following figure.

Figure 63. Logical And of Reference Clock Signals



  - An exception is made in the case of PCIe, where PCIe must have its system PLLs reference clock ready but the other reference clocks for the other system PLLs are not ready. In this case, the transceiver is configured for PCIe operation prior to other system PLLs and transceivers being up and running. All other system PLLs need to wait until every reference clock for all system PLLs are ready.
- Once the reference clock for the system PLL is up; it must be stable; it must be present throughout the device operation and must not go down. If you are not able to adhere to this, you must reconfigure the device. After the temporary loss of the system PLL reference clock, you may observe that the first try of device reconfiguration fails. If that happens, you should try to reconfigure the device a second time.

4.5. Guidelines to Indicate System PLL Reference Clock is Ready

4.5.1. Example flow to indicate System PLL reference clock is ready

Here are the steps to indicate that the system PLL reference clock is ready.

If reference clock is ready before device configuration:

1. Tie `i_refclk_ready` pin to high.

If reference clock is not ready before device configuration:

1. Wait until the system PLL's reference clock is available and stable.
2. Set `i_refclk_ready` pin to high.

Note: For PCIe you must select one of the PCIe modes in the GTS System PLL Clocks Intel FPGA IP. When you select PCIe mode, the `i_refclk_ready` port is not available. You must make sure that the reference clock to the system PLL is available and stable before device configuration.

5. Implementing the GTS Reset Sequencer Intel FPGA IP

The following chapter describes the implementation of the GTS Reset Sequencer Intel FPGA IP. Refer to the chapter for implementation details of IP instantiation and connections for Intel Agilex 5 designs.

This is a mandatory IP and must be instantiated for simulation and proper device operation of the Intel Agilex 5 FPGAs.

5.1. IP Requirements

The GTS Reset Sequencer Intel FPGA IP must be instantiated for each side of the device that uses transceivers. Refer to the *Transceiver Architecture* chapter for more information. Based on your design you must instantiate one or two of instances of the IP:

- One GTS Reset Sequencer Intel IP instance if your design uses transceivers on one side of the device.
- Two GTS Reset Sequencer Intel IP instances if your design uses transceivers on both sides of the device.

The following table shows the logic usage in the FPGA fabric of the GTS Reset Sequencer Intel FPGA IP.

Table 59. Logic Usage of the GTS Reset Sequencer Intel FPGA IP (For Intel Agilex 5 E-Series and D-Series Devices)

Device Family	ALM[25]	ALUT	Logic Register	M20K
Intel Agilex 5 E-Series (12 Lanes)	88	107	113	0
Intel Agilex 5 D-Series (16 Lanes)	108	122	137	0

5.2. IP Parameters

The table below lists the IP parameters for the GTS Reset Sequencer Intel FPGA IP.

(25) Logic utilization is lower for fewer channel applications.

Figure 64. GTS Reset Sequencer Intel FPGA IP Parameter Editor

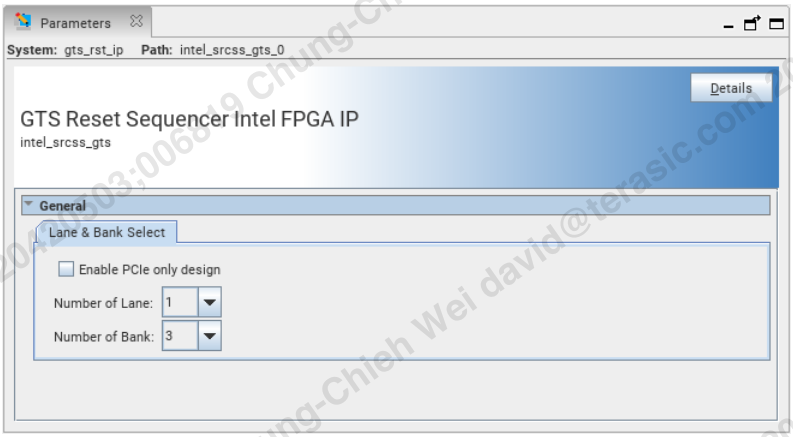


Table 60. GTS Reset Sequencer Intel FPGA IP Parameters

Parameter Name	Default	Range	Description
Enable PCIe only design	**Off**	**On/Off**	Enable or disable the PCIe only design setting per GTS Reset Sequencer Intel FPGA IP. Enable – Only the `pma_cu_clk` port is available from the GTS Reset Sequencer Intel FPGA IP. Disable – All ports are available from the GTS Reset Sequencer Intel FPGA IP. *Note:* Enable this feature if you are using PCIe only or USB3.1 only or combination of both per side of the device.
Number of Lane	**1**	**1 - 16**	Number of lanes per side of the device to be connected to the GTS Reset Sequencer Intel FPGA IP. *Note:* The **Number of Lane** parameter must be set to the exact number of reset sequencer request or grant signals used and you cannot leave them unconnected. The **Number of Lane** parameter does not include PCIe and USB3.1 channels.
Number of Bank	**3**	**1 - 4**	Number of banks per GTS Reset Sequencer Intel FPGA IP. *Note:* The number of banks reflects the maximum allowable number of lanes in your design. The **Number of Banks** parameter must be set to the exact number of transceiver banks used in your design and cannot be left unconnected.

5.3. IP Port List

Table 61. GTS Reset Sequencer Intel FPGA IP Port List

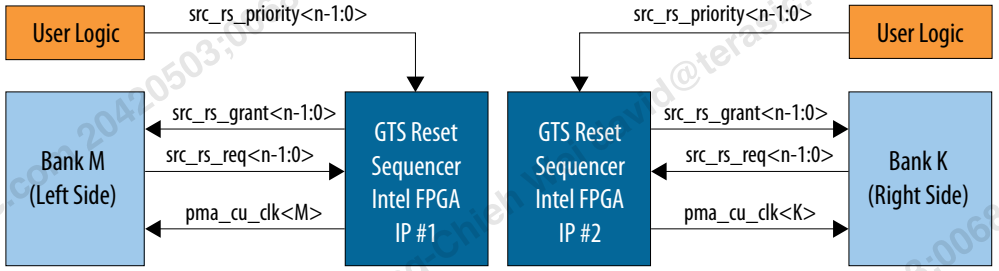
N is number of channels used.

M is number of banks per side of the device.

Signal Name	Direction	Width	Description
i_src_rs_req	Input	N	Request from SRC to GTS Reset Sequencer Intel FPGA IP for reset operation. Asserts when there is a request to toggle reset.
o_src_rs_grant	Output	N	Grant from GTS Reset Sequencer Intel FPGA IP to SRC. Asserts when the reset request is granted by the Reset Sequencer.
i_src_rs_priority	Input	N	Binary priority input • 0 - Low priority • 1 - High priority This port used to set priority for a channel that you need to prioritize the reset sequence when there are multiple channels being reset simultaneously. You must tie the input to 0 except for the priority channel which needs to be set to 1.
o_pma_cu_clk	Output	M	PMA control unit clock output, one per bank for each side of the device. This clock port must be connected to the GTS PMA/FEC Direct PHY Intel FPGA IP and all other protocol IPs.

5.4. GTS Reset Sequencer Intel FPGA IP General Interface

Figure 65. GTS Reset Sequencer Intel FPGA IP General Interface



Legends:
M = Count of Left Side Banks for the device
K = Count of Right Side Banks for the device
n = Total number of Lanes

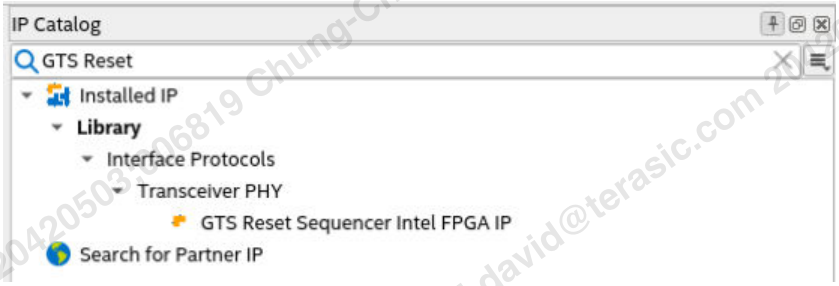
Notes:
Each bank will have maximum 4 channels. All the ports must not cross side.

5.5. GTS Reset Sequencer Intel FPGA IP Design Flow

The design flow for the GTS Reset Sequencer Intel FPGA IP is described below:

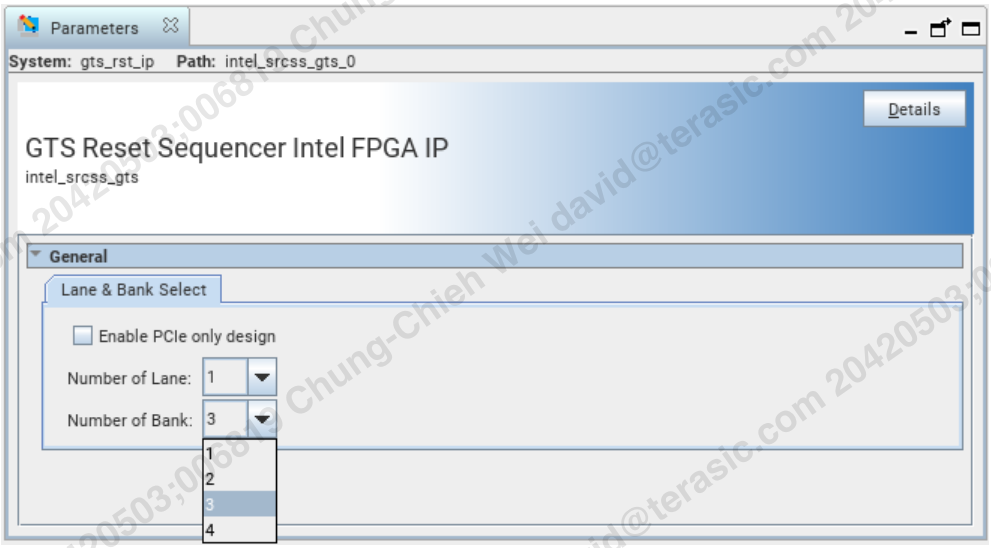
1. Add the GTS Reset Sequencer Intel FPGA IP from the **IP Catalog** into your design as shown in the following figure.

Figure 66. IP Catalog



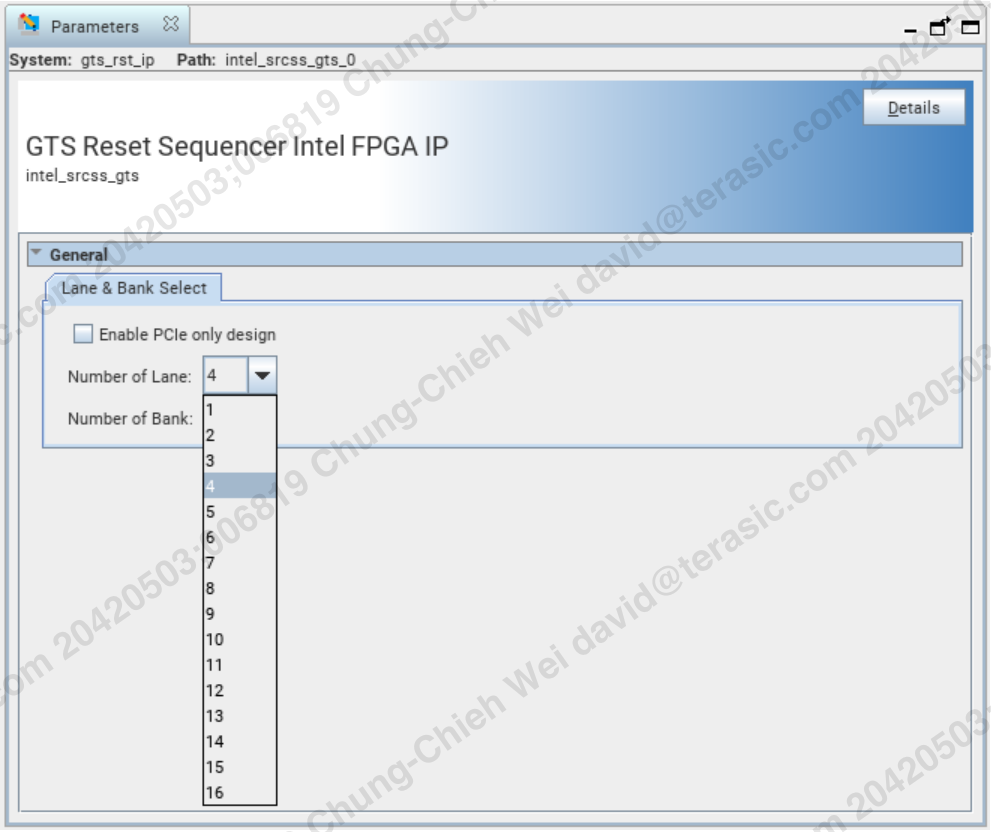
2. Select the total **Number of Banks** that you want to use for the GTS Reset Sequencer Intel FPGA IP as shown in the following figure.

Figure 67. IP Parameters for Bank Selection



3. Select the total **Number of Lanes** that you want to use for the GTS Reset Sequencer Intel FPGA IP as shown in the following figure.

Figure 68. IP Parameters for Lane Selection



4. Connect `o_src_rs_grant` and `i_src_rs_req` to the channels. The `o_src_rs_grant` and `i_src_rs_req` must be connected to the same channel so that the reset operation works accordingly.
5. Connect `o_pma_cu_clk` to `i_pma_cu_clk` input of the GTS PMA/FEC Direct PHY Intel FPGA IP and protocol IPs. If there are two or more IPs in the same bank, the IPs must be connected to the same `o_pma_cu_clk`. For any separate bank, make sure to use different `o_pma_cu_clk` for each bank.
6. For channels that need to be prioritized for reset sequencing, tie `i_src_rs_priority` to `1` for that specific channel based on the connection of bits `o_src_rs_grant` and `i_src_rs_req` for that channel. For non-priority (normal) reset sequence channels, tie the `i_src_rs_priority` to `0`. For example, the value `4'b0010`, sets the priority to lane 2.

Note: You can skip steps 3. 4, and 6. if you set the **Enable PCIe only design** option to **Enable** in the IP parameter GUI. If you are using Platform Designer, for steps 4 and 5, you must connect `o_src_rs_grant`, `i_src_rs_req`, and `o_pma_cu_clk` signals using wire-level expressions. Refer Editing Wire-Level Expressions in the Intel Quartus Prime Pro Edition User Guide: Platform Designer for more details.

5.6. GTS Reset Sequencer Intel FPGA IP Use Cases

5.6.1. Example Use Case 1

In this example use case, both sides of the device are fully populated and have the following IPs instantiated:

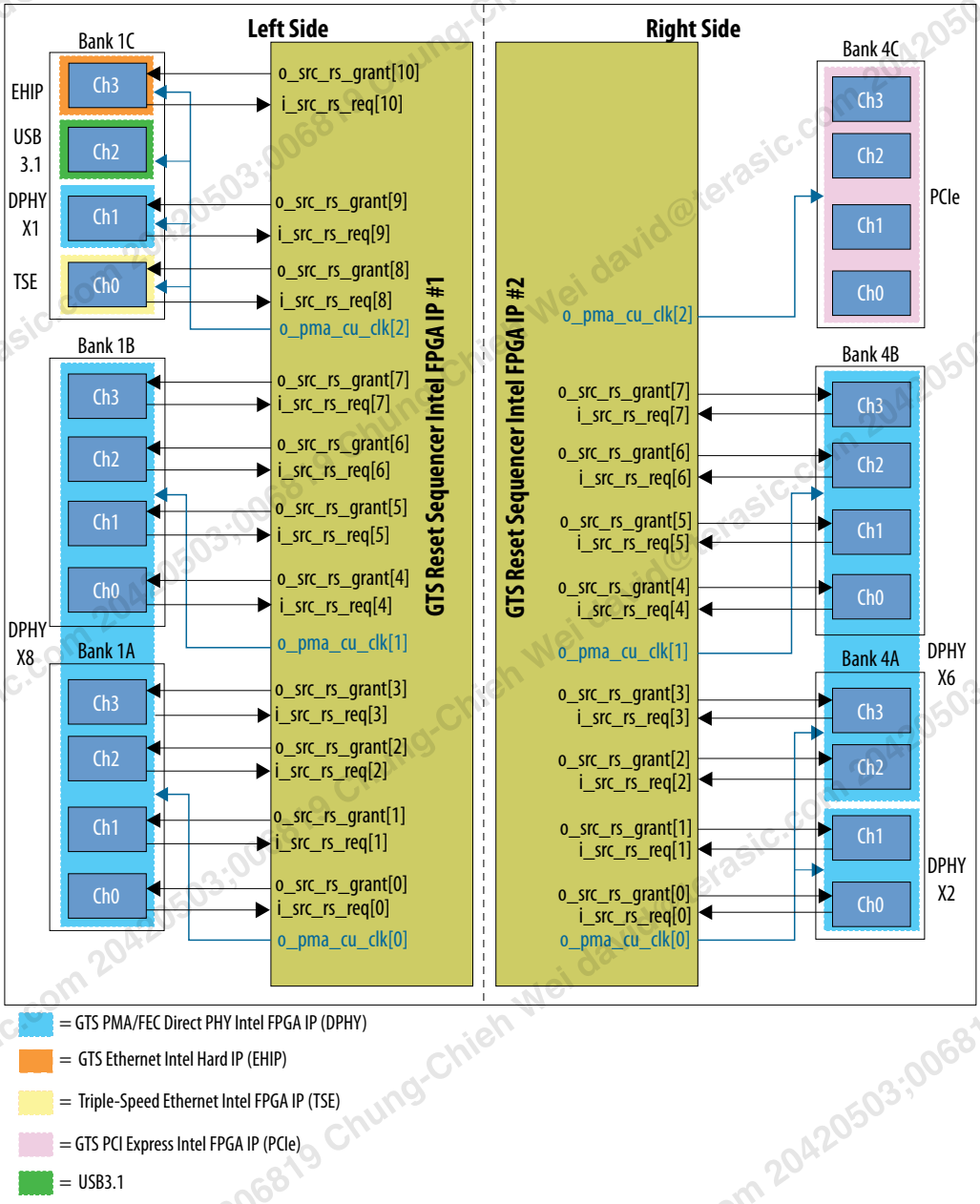
- Two GTS Reset Sequencer Intel FPGA IP
- Four GTS PMA/FEC Direct PHY Intel FPGA IP
- One GTS Ethernet Intel FPGA Hard IP
- One GTS PCI Express Intel FPGA IP
- One Triple-Speed Ethernet Intel FPGA IP
- One USB 3.1

Table 62. GTS Reset Sequencer Intel FPGA IP Parameter Settings for Use Case 1

GTS Reset Sequencer Intel FPGA IP	Parameter	Value Selection
# 1 (Left Side)	**Enable PCIe only design**	**Off**
	Number of Lane	**11**
	Number of Bank	**3**
# 2 (Right Side)	**Enable PCIe only design**	**Off**
	Number of Lane	**8**
	Number of Bank	**3**

The following figure shows the connections between the two GTS Reset Sequencer Intel FPGA IPs and the other instantiated IPs.

Figure 69. Example Use Case 1



5.6.2. Example Use Case 2

In this example use case, both sides of the device are not fully populated and have the following IPs instantiated:

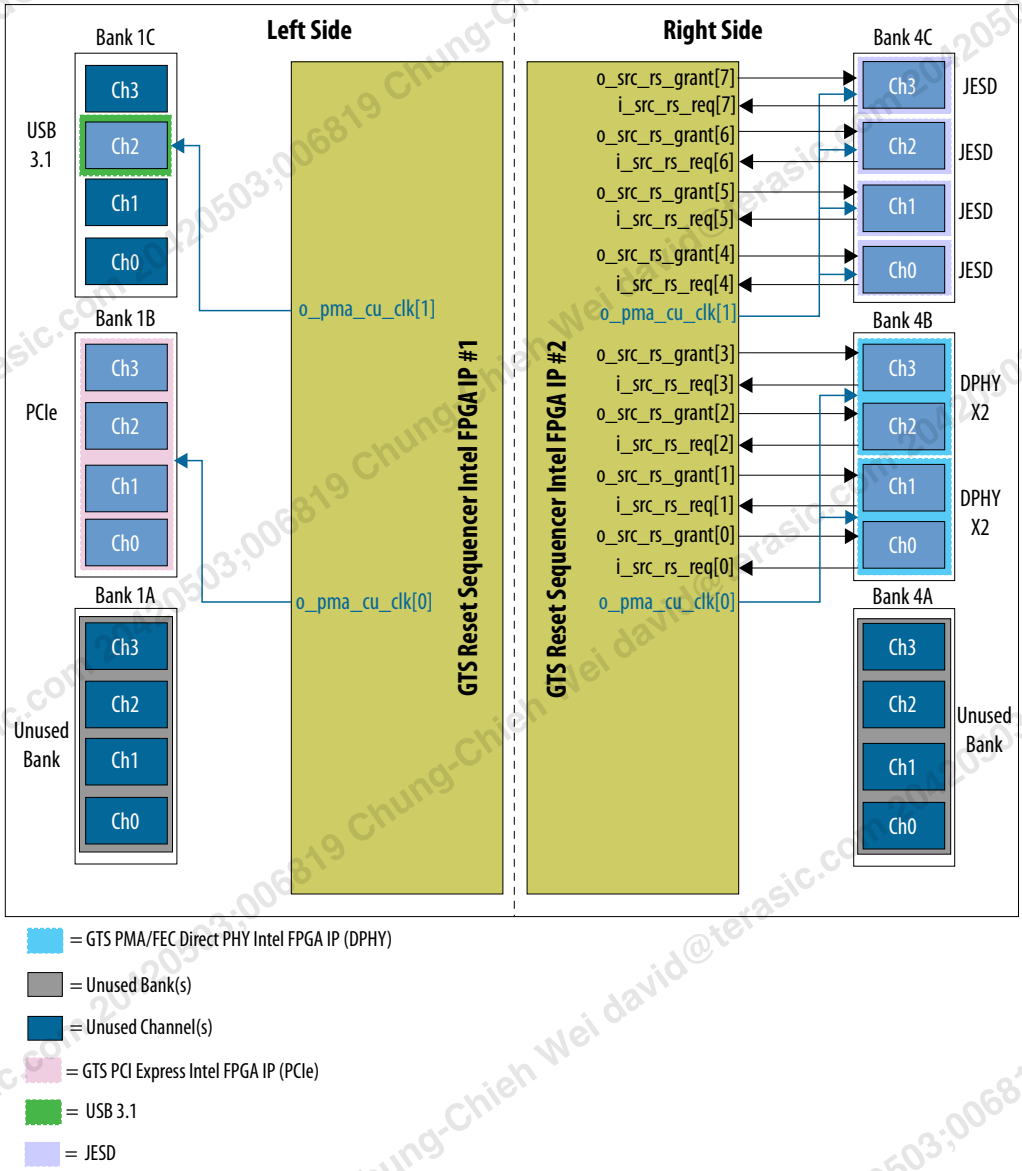
- Two GTS Reset Sequencer Intel FPGA IP
- Two GTS PMA/FEC Direct PHY Intel FPGA IP
- Four JESD204C GTS Intel FPGA IP
- One GTS PCI Express Intel FPGA IP
- One USB 3.1

Table 63. GTS Reset Sequencer Intel FPGA IP Parameter Settings for Use Case 2

GTS Reset Sequencer Intel FPGA IP	Parameter	Value Selection
# 1 (Left Side)	**Enable PCIe only design**	**On**
	Number of Lane	**—**
	Number of Bank	**2**
# 2 (Right Side)	**Enable PCIe only design**	**Off**
	Number of Lane	**8**
	Number of Bank	**2**

The use case also shows you how to use the **Enable PCIe only design** parameter in a design. In this use case, on the left side of the device, the `i_src_rs_req` and `o_src_rs_grant` ports are not needed but `o_pma_cu_clk` is still needed and cannot be left unconnected..

Figure 70. Example Use Case 2



6. GTS PMA/FEC Direct PHY Intel FPGA IP Example Design

This chapter describes the **Example Design** generation in the GTS PMA/FEC Direct PHY Intel FPGA IP. There are a few example designs supported currently and these example designs show the various connections between the IPs and their configuration. The following IPs from the Intel Quartus Prime Pro Edition software IP catalog are used in all the example designs:

- GTS PMA/FEC Direct PHY Intel FPGA IP
- GTS System PLL Clocks Intel FPGA IP
- GTS Reset Sequencer Intel FPGA IP

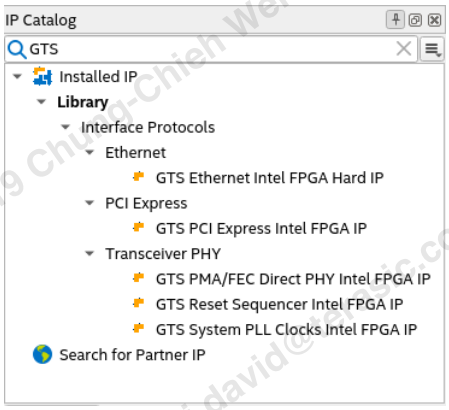
The example design also provides a simulation testbench that supports compilation and simulation. When you generate the example design, the parameter editor automatically creates the files necessary to simulate the design. You can use the supported simulator to run the testbench to observe the GTS PMA/FEC Direct PHY Intel FPGA IP functional simulation results and behavior.

6.1. Instantiating the GTS PMA/FEC Direct PHY Intel FPGA IP

To instantiate the GTS PMA/FEC Direct PHY Intel FPGA IP:

1. Specify the target device family, click **Assignments ➤ Device**, and then select **Agilex A5ED065BB32AE5SR0**.
2. If IP catalog is not already open, click **View ➤ IP Catalog** in the Intel Quartus Prime Pro Edition software.
3. In the IP Catalog search field, type GTS PMA, and double-click the **GTS PMA/FEC Direct PHY Intel FPGA IP**.

Figure 71. IP Catalog



6.2. Generating the GTS PMA/FEC Direct PHY Intel FPGA IP Example Design

To generate the example design you need to open the GTS PMA/FEC Direct PHY Intel FPGA IP and go to the **Example Design** tab. The GTS PMA/FEC Direct PHY Intel FPGA IP parameter editor includes the **Generate Example Design** function to easily create, generate, and simulate a GTS PMA or FEC direct mode example design.

You can currently select one out of the five **Example Design Options** for generation as shown in the following table.

Table 64. Example Design Options

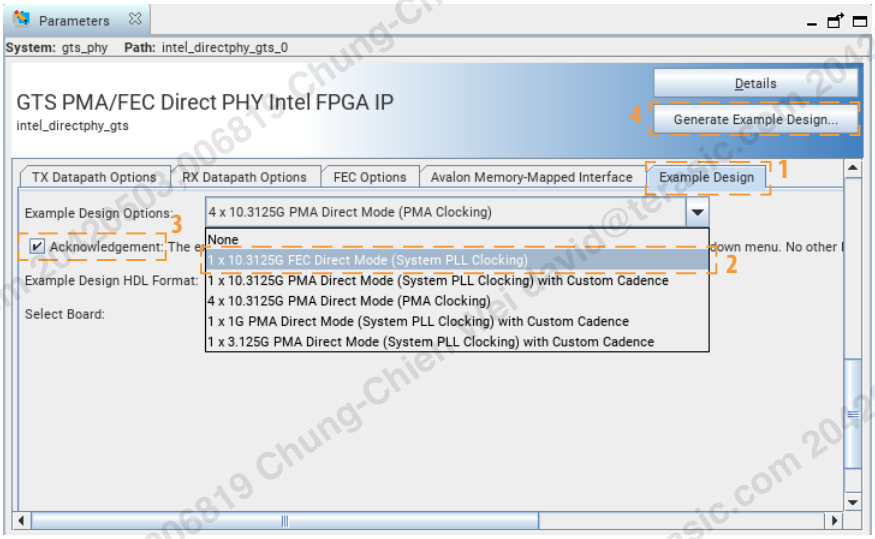
Example Design Options	Description
1 x 10.3125G FEC Direct Mode (System PLL Clocking)	One NRZ FEC Direct GTS lane, with a throughput of 10.3125 Gbps, with System PLL clocking mode
1 x 10.3125G PMA Direct Mode (System PLL Clocking) with Custom Cadence	One NRZ PMA Direct GTS lane, with a throughput of 10.3125 Gbps, with System PLL clocking mode and custom cadencing
4 x 10.3125G PMA Direct Mode (PMA Clocking)	Four NRZ PMA Direct GTS lane, with 10.3125 Gbps per PMA lane, with PMA clocking mode
1 x 1G PMA Direct Mode (System PLL Clocking) with Custom Cadence	One NRZ PMA Direct GTS lane, with a throughput of 1Gbps, with System PLL clocking mode and custom cadencing
1x 3.125G PMA Direct Mode (System PLL Clocking) with Custom Cadence	One NRZ PMA Direct GTS lane, with a throughput of 3.125 Gbps, with System PLL clocking mode and custom cadencing.

To generate an example design, follow the steps below:

1. Go to the **Example Design** tab in the GTS PMA/FEC Direct PHY Intel FPGA IP.
2. Select one of the example designs from the drop-down menu. If you select **None** you cannot generate the example design.
3. Click the **Acknowledgment** option box. This option is to remind you that only the example design you specify in the drop-down menu is generated. If you make any modification to the parameter settings of the IP after selecting the **Example Design** options from the drop down list, the changes you make to the IP parameters do not take effect. Only the parameters defined for the **Example Design** options in Example Design Options table take effect. If you do not check the acknowledgment box, you cannot generate the example design.
4. Ensure steps 2. and step 3. are done, then click on **Generate Example Design**. Clicking **Generate Example Design** completes the **IP Generation**. An example design folder is generated containing the Intel Quartus Prime project (.qpf), settings (.qsf), and IP files. In addition, there are two folders created named `rtl` and `testbench` containing the RTL and simulation testbench files in the following location:

```
<Project Folder>/<directphy_example_design/example_design>
```

Figure 72. GTS PMA/FEC Direct PHY Intel FPGA IP Example Design Steps



Note: If you select any of the five available **Example Design Options**, but change the GTS PMA/FEC Direct PHY Intel FPGA IP settings in the GUI thereafter, the example design generated does not follow the changed settings for the GTS PMA/FEC Direct PHY Intel FPGA IP. The example design generation only takes the **Example Design Options** listed in **Example Design** tab of the **IP Parameter** editor. Any other changes that you make to the GTS PMA/FEC Direct PHY Intel FPGA IP settings are not applied during example design generation.

6.2.1. GTS PMA/FEC Direct PHY Intel FPGA IP Example Design Directory Structure

The example design RTL files are in the `example_design/rtl` directory and simulation files are in the `example_design/testbench` directory. The GTS PMA/FEC Direct PHY Intel FPGA IP example design generates the following files:

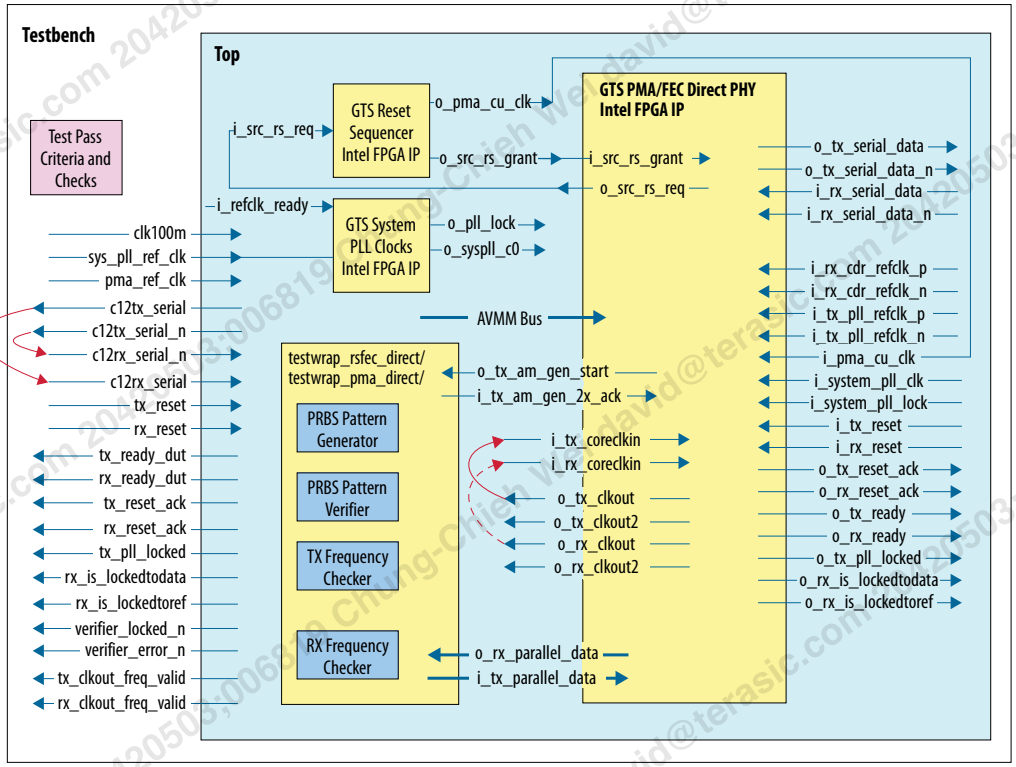
Table 65. Example Design Directory Structure and Description

Directory Structure and File Name	Description
Key testbench and simulation files for the GTS PMA/FEC Direct PHY Intel FPGA IP Example Designs	
`<directphy_example_design>/example_design/rtl/top_tst.sv`	Top-level testbench file. The testbench instantiates the `top.v` PMA direct design file.
`<directphy_example_design>/example_design/rtl/testwrap_pma_direct.sv`	Test wrapper file that generates and receives the PRBS data stream as well as performs the TX and RX clock output frequency checks.
Testbench scripts for the GTS PMA/FEC Direct PHY Intel FPGA IP Example Designs	
`<directphy_example_design>/example_design/testbench/run_vcs.sh`	The VCS script to run the testbench.
`<directphy_example_design>/example_design/testbench/run_vsim.tcl`	The QuestaSim script to run the testbench.

6.3. GTS PMA/FEC Direct PHY Intel FPGA IP Example Design Functional Description

The GTS PMA/FEC Direct PHY Intel FPGA IP example design simulation testbench top-level block diagram is shown in the following figure.

Figure 73. Simulation Testbench Block Diagram for the GTS PMA/FEC Direct PHY Intel Example Design



This section provides the functional description of the example design and the simulation results for both the PMA and FEC direct designs listed in the following table.

Table 66. Example Design Functional Description

Example Design Option	Functional Description
1 x 10.3125G FEC Direct Mode (System PLL Clocking)	One NRZ FEC Direct GTS lane operating at 10.3125 Gbps with System PLL clocking mode
4 x 10.3125G PMA Direct Mode (PMA Clocking)	Four NRZ PMA Direct GTS lane operating at 10.3125 Gbps per PMA lane with PMA clocking mode

The testbench program controls the testbench components through the Avalon memory-mapped interface. For both the PMA and FEC direct example designs, the testwrap block consists of the PRBS generator, PRBS verifier, and TX and RX clock output frequency checkers. There are 2 types of test wrap blocks:

- PMA test wrap – used in PMA direct configurations.
- FEC test wrap – used in FEC direct configuration.

The clock sources for the example design are shown in the following table.

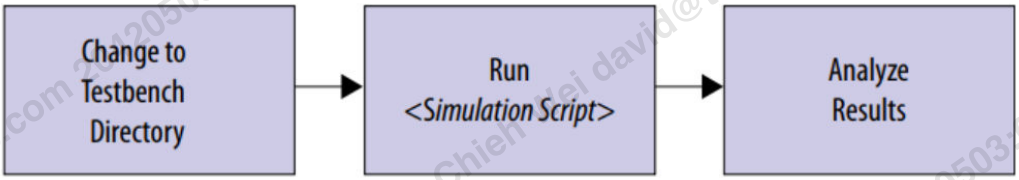
Table 67. Example Design Clock Sources

Example Design Option	Clock Source Connections
1 x 10.3125G FEC Direct Mode (System PLL Clocking)	• 100 MHz for reconfiguration clock • 156.25 MHz reference clock for the system PLL • 156.25 MHz for the GTS PMA direct channel as TX PLL and RX CDR reference clock
4 x 10.3125G PMA Direct Mode (PMA Clocking)	• 100 MHz for testbench reset logic, management clock, and reconfiguration clock • 156.25 MHz for the GTS PMA direct channel as TX PLL and RX CDR reference clock

6.4. Simulating the GTS PMA/FEC Direct PHY Intel FPGA IP Example Design Testbench

Follow these steps to simulate the testbench:

Figure 74. Steps to Simulate the Example Design



1. At the command prompt, change to the testbench simulation directory `<example_design/testbench>`.

```
cd <directphy_example_design>/example_design/testbench
```

2. Run the simulation using the supported simulators by executing the simulation script file. To simulate with VCS, change to the `example_design/testbench` directory and the launch the simulation using the shell script:

```
sh run_vcs.sh
```

To run the simulation in QuestaSim, run the following command:

```
vsim -c -do run_vsim.tcl
```

Note: Currently only the VCS and QuestaSim simulators are supported

3. The following steps show the simulation testbench flow for the example design:
   a. Assert resets `i_tx_reset` and `i_rx_reset` to reset the IP.
   b. Wait until resets are acknowledged, when `o_tx_reset_ack` and `o_rx_reset_ack` go high.
   c. Deassert the resets, `i_tx_reset` and `i_rx_reset`. Monitor `o_tx_ready` bit is set to 1, indicating TX path is ready.
   d. Monitor `o_rx_ready` bit is set to 1, indicating the RX path is ready.
   e. Monitor `o_tx_pll_locked` bit is set to 1, indicating that the TX PLL is locked to reference clock within the PPM threshold status signal.

f. Monitor `o_rx_is_lockedtoref` bit is set to 1, indicating the CDR is frequency locked to reference clock within the PPM threshold.

g. Monitor `o_rx_is_lockedtodata` bit is set to 1, indicating indicates that the CDR is in locked-to-data mode.

h. Monitor `tx_clkout_freq_valid` bit is set to 1, indicating TX clock output frequency is within the upper and lower limits as expected in the definition file.

i. Monitor `rx_clkout_freq_valid` bit is set to 1, indicating RX clock output frequency is within the upper and lower limits as expected in the definition file.

j. Monitor `verifier_lock` bit is set to 1, indicating that the lock to the RX data pattern after successfully predicting 16 consecutive patterns in RX data.

k. Monitor `verifier_error` bit is not set to 1. If it is 1, this indicates the RX data is different than the expected result.

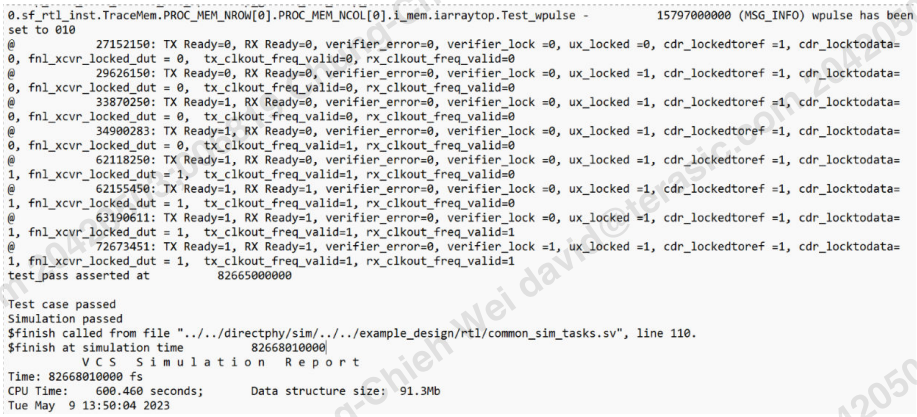
4. Analyze the results, a passing testbench displays the following messages in the simulation window, `Test case Passed` and `Simulation Passed`, as shown in the following figures.

Figure 75. Sample Results for the PMA Direct PHY Example Design Testbench

```
0, ux_locked =0, cdr_lockedtoref =1, cdr_locktodata= 0, fnl_xcvr_locked_dut = 0,
  tx_clkout_freq_valid=0, rx_clkout_freq_valid=0
@            29626150: TX Ready=0, RX Ready=0, verifier_error=0, verifier_lock =
0, ux_locked =1, cdr_lockedtoref =1, cdr_locktodata= 0, fnl_xcvr_locked_dut = 0,
  tx_clkout_freq_valid=0, rx_clkout_freq_valid=0
@            29693450: TX Ready=1, RX Ready=0, verifier_error=0, verifier_lock =
0, ux_locked =1, cdr_lockedtoref =1, cdr_locktodata= 0, fnl_xcvr_locked_dut = 0,
  tx_clkout_freq_valid=0, rx_clkout_freq_valid=0
@            30723591: TX Ready=1, RX Ready=0, verifier_error=0, verifier_lock =
0, ux_locked =1, cdr_lockedtoref =1, cdr_locktodata= 0, fnl_xcvr_locked_dut = 0,
  tx_clkout_freq_valid=1, rx_clkout_freq_valid=0
@            62098250: TX Ready=1, RX Ready=0, verifier_error=0, verifier_lock =
0, ux_locked =1, cdr_lockedtoref =1, cdr_locktodata= 1, fnl_xcvr_locked_dut = 1,
  tx_clkout_freq_valid=1, rx_clkout_freq_valid=0
@            62135450: TX Ready=1, RX Ready=1, verifier_error=0, verifier_lock =
0, ux_locked =1, cdr_lockedtoref =1, cdr_locktodata= 1, fnl_xcvr_locked_dut = 1,
  tx_clkout_freq_valid=1, rx_clkout_freq_valid=0
@            63131204: TX Ready=1, RX Ready=1, verifier_error=0, verifier_lock =
1, ux_locked =1, cdr_lockedtoref =1, cdr_locktodata= 1, fnl_xcvr_locked_dut = 1,
  tx_clkout_freq_valid=1, rx_clkout_freq_valid=0
@            63171983: TX Ready=1, RX Ready=1, verifier_error=0, verifier_lock =
1, ux_locked =1, cdr_lockedtoref =1, cdr_locktodata= 1, fnl_xcvr_locked_dut = 1,
  tx_clkout_freq_valid=1, rx_clkout_freq_valid=1
test_pass asserted at          73125000000

Test case passed
Simulation passed
$finish called from file "../../directphy/sim/../../example_design/rtl/common_si
m_tasks.sv", line 110.
$finish at simulation time          73128010000
           V C S   S i m u l a t i o n   R e p o r t
Time: 73128010000 fs
CPU Time:    577.390 seconds;       Data structure size: 136.3Mb
Mon May 22 15:30:33 2023
```

Figure 76. Sample Results for the FEC Direct PHY Example Design Testbench



6.4.1. Modifying the Example Design and Performing Simulation

If you wish to modify the example design to change the data rate, system PLL clock frequency, increase the number of PMA lanes and so on, you can reuse the existing example design and perform following changes:

1. Update and re-configure the GTS PMA/FEC Direct PHY Intel FPGA IP, GTS System PLL Clock Intel FPGA IP, and GTS Reset Sequencer Intel FPGA IP.
   a. Generate and instantiate the GTS Reset Sequencer Intel FPGA IP and make sure the connections of the `i_src_rs_req` and `o_src_rs_grant` ports are connected correctly to the GTS PMA/FEC Direct PHY Intel FPGA IP. If you add more GTS transceiver banks in the design, you must ensure proper connections for the `o_pma_cu_clk` port. Refer to Implementing the GTS Reset Sequencer Intel FPGA IP for more information.

   Note: You must ensure that the system PLL frequency in the GTS PMA/FEC Direct PHY Intel FPGA IP and GTS System PLL Clocks Intel FPGA IP is set to the same value, if you are using the system PLL clocking mode.
2. Regenerate the IPs by clicking **Generate HDL**.
3. Run **Analysis and Synthesis**.
4. Initialize and make changes to the testbench variable files in the following example design directory <example_design/rtl>:
   a. `testwrap_pma_direct.sv` and `test_tst.sv`
   b. `param_defines.iv` and `param_defines1.iv`
5. After making the necessary changes, refer to *Simulating the GTS PMA/FEC Direct PHY Intel FPGA IP Example Design Testbench* to run the simulation and analyze results.

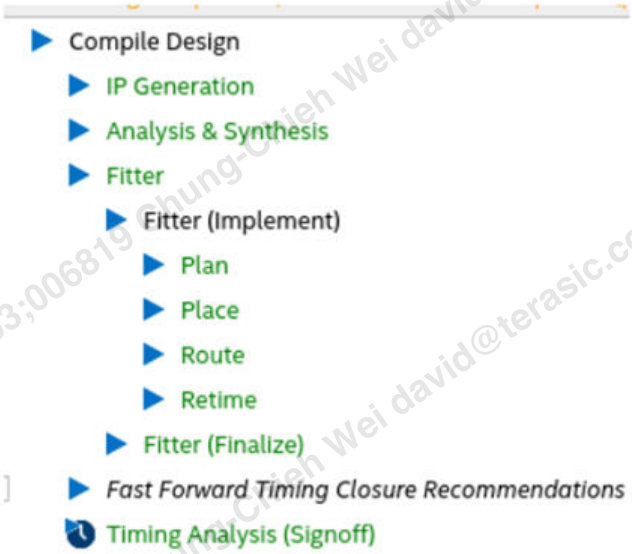
Note: For a successful simulation, you must ensure that the **Refclk frequency** of the GTS System PLL Clocks Intel FPGA IP and **RX CDR reference clock frequency** together with **TX PLL Integer mode reference clock frequency** in the GTS PMA/FEC Direct PHY Intel FPGA IP are driven from the same clock source.

6.5. Compiling the GTS PMA/FEC Direct PHY Intel FPGA IP Example Design

For successful compilation, you must ensure the following connections are made correctly:

1. Instantiate all the IPs below in the top level file.
   - GTS PMA/FEC Direct PHY Intel FPGA IP
   - GTS System PLL Clocks Intel FPGA IP
   - GTS Reset Sequencer Intel FPGA IP
2. Connect port `i_refclk` of the GTS System PLL Clocks Intel FPGA IP to the port `i_rx_cdr_refclk` and `i_tx_pll_refclk` of the GTS PMA/FEC Direct PHY Intel FPGA IP. You must also ensure the source of the reference clock is coming from the same clock.
3. Connect port `o_pma_cu_clk` of the GTS Reset Sequencer Intel FPGA IP to port `i_pma_cu_clk` of the GTS PMA/FEC Direct PHY Intel FPGA IP.
4. Run all stages as shown in the figure below:

Figure 77. Example Design Compilation Flow



7. Design Assistance Tools

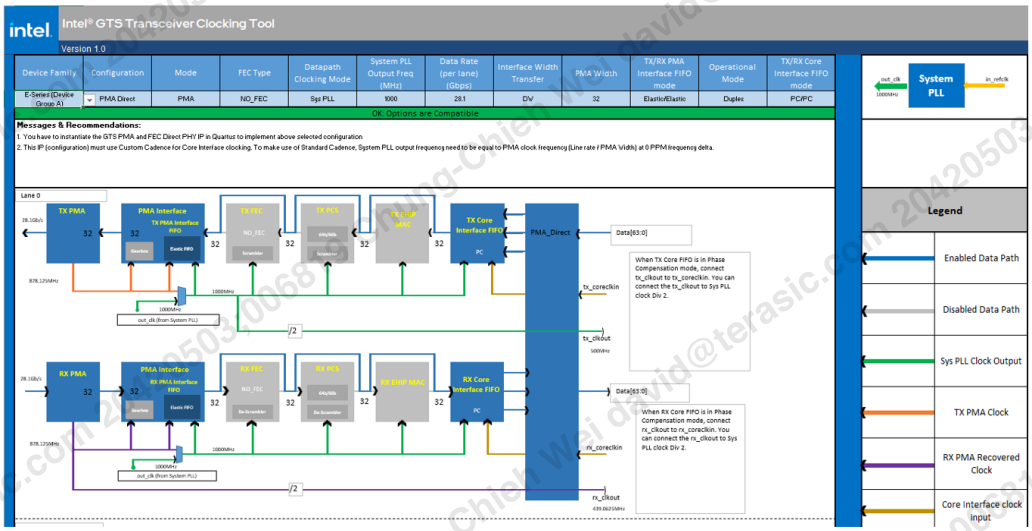
7.1. Clocking and Datapath Tool

The *Clocking and Datapath Tool* helps you to explore and plan your clocking resources for a GTS transceiver design and displays the following:

- TX and RX datapath of one configuration at a time.
- Selection of parameters that can affect clocking.
- Parameters such as PMA and FEC Direct modes, Ethernet mode, FEC type, clocking scheme, PMA width, and core and interface FIFO.

The Microsoft Excel based *Clocking and Datapath Tool* is available for download at *Clocking and Datapath Tool*. A screenshot of the tool is shown in the following figure.

Figure 78. Clocking and Datapath Tool



Note: The current release of Intel Quartus Prime Pro Edition software has support for limited clock frequencies. Refer to the *Implementing the GTS System PLL Clock Intel FPGA IP* to confirm support for frequencies shown in the *Clocking and Datapath tool*.

7.2. TX Equalizer Tool

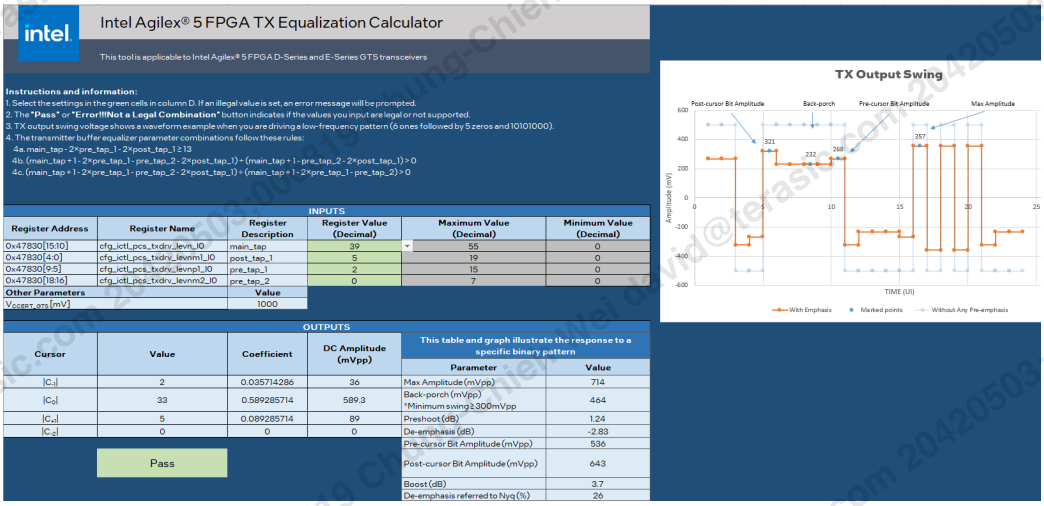
A high-speed signal traveling through a backplane is subject to high-frequency losses, primarily skin effect and dielectric losses. These losses can severely degrade and attenuate the high-frequency content of the signal, making it difficult for the receiver to interpret the signal. The Intel Agilex 5 transmitter PMA offers pre-emphasis and linear equalization to address this problem and improve the high-speed signal quality.

You can use the Intel Agilex 5 TX Equalizer Tool to estimate the pre-emphasis and output swing. The tool displays the Intel Agilex 5 PMA transmitter output swing when you change the transmitter's:

- Main_tap
- Post_tap_1
- Pre_tap_1
- Pre_tap_2

The Excel-based Intel Agilex 5 TX Equalizer Tool, is available for download at *Intel Agilex 5 TX Equalizer Tool*. A screenshot of the tool is shown in the following figure.

Figure 79. Intel Agilex 5 TX Equalizer Tool



8. Debugging GTS PMA Transceiver Links with Transceiver Toolkit

The GTS PMA/FEC Direct PHY Intel FPGA IP supports the Transceiver Toolkit as part of debugging tools that you can use to debug your GTS Transceiver Links. The Transceiver Toolkit helps you to optimize high-speed serial links in your board design by providing real-time control, monitoring, and debugging of the GTS transceiver links running on your board. The Transceiver Toolkit support is planned to be available in a future Intel Quartus Prime Pro Edition software release.

The Transceiver Toolkit features allow you to:

- Test and tune transceiver link signal quality through a combination of metrics.
- Run a single bit-error-rate (BER) test on a single channel or multiple simultaneous BER tests on multiple channels.
- Use Auto Sweep feature to sweep transceiver settings to determine the parameters that support best BER value.
- Generate and check different PRBS patterns to provides you the ability to stress your link according to your system specifications.
- View the receiver horizontal and vertical eye margin during testing.

Table 68. Transceiver Toolkit GUI Parameters that are Planned (Preliminary)

Parameter	Description
Auto Sweep Status	Reports the current and best tested bits, errors, bit error rate, and case count for the current Auto Sweep test.
Bit error rate (BER)	Reports the number of errors divided by bits tested since the last reset of the checker. When RX CDR is locked to reference clock or PRBS checker is not locked, the BER reported is not reliable.
Loopback Mode	• TX2RXBUF – Loops back the TX serializer output to the RX Equalizer input. • TX2RXPAR – Loops back the parallel data from TX to the parallel RX input of the transceiver. • RX2TXPAR – Loops back the RX signal output from the transceiver (parallel output) to the parallel TX input of the transceiver.
PRBS Pattern	Allows you to select the PRBS pattern for the bit error test. Available PRBS pattern are PRBS31, PRBS28, PRBS23, PRBS15, PRBS13, PRBS10, PRBS9, PRBS7, SSPR, SSPR1.
RX PMA Settings	Displays the values of the High Frequency VGA Gain, High Frequency Boost, and DFE Data Tap 1.
RX PMA Advance Settings	Displays the values of the DFE Data Taps 2 through 16.
TX EQ Parameters	Allows you to control the TX Equalization parameters such as pre_tap_2, pre_tap_1, and post_tap_1.

continued...

Parameter	Description
Eye Measurements	Provides interface for measuring the Eye Width and Eye Height as well as saving the measurements into a CSV file.
Set Working Directory	Allows you to change the folder used as the working directory. You can use this setting to save the exported eye measurements CSV file to your chosen path.
Invert Polarity	Allows you to enable TX or RX polarity inversion
RX CDR Lock Status	Display the RX CDR status. There are two lock status indicators, **RX CDR locked to data** and **RX CDR lock to ref clock**.

9. Document Revision History for the Intel Agilex 5 FPGA GTS Transceiver Architecture and PMA and FEC Direct PHY IP User Guide

Document Version	Intel Quartus Prime Version	Changes
2023.12.18	23.4	Made the following changes: • Added additional information to implement power down for unused banks in the *Unused PMA rules* section. • Added *Preset IP Parameter Settings* section. • Updated parameters in the *General and Common Datapath Options*, *TX Datapath Options* and *RX Datapath Options* sections. • Added *Core PLL Mode* section. • Added *PMA and FEC Direct PHY Soft CSR Registers* and *GTS PMA Register Map* sections. • Added *Configurable Intel Quartus Prime Software Settings* section. • Added *Configuring the GTS PMA/FEC Direct PHY FPGA Intel IP for Hardware Testing* section. • Updated the *IP Parameters* and *Mode of System PLL - System PLL Reference Clock and Output Frequencies* sections in the *Implementing the GTS System PLL Clocks Intel FPGA IP* chapter. • Updated the *IP Requirements*, *IP Parameters*, *GTS Reset Sequencer Intel FPGA IP Design Flow*, and *GTS Reset Sequencer FPGA IP Use Cases* sections in the *Implementing the GTS Reset Sequencer Intel FPGA IP* chapter. • Added new chapter *Debugging GTS PMA Transceiver Links with Transceiver Toolkit*.
2023.08.11	23.2.1	Made the following changes: • Added information about Intel Agilex 5 D-Series devices in the *GTS Transceiver Overview* and *GTS Transceiver Architecture* chapters. • Made several updates in the *Clock Architecture* and *FEC Architecture* sections of the *GTS Transceiver Architecture* chapter. • Added the following new chapters: — *Implementing the GTS PMA/FEC Direct PHY Intel FPGA IP* — *Implementing the GTS System PLL Clock Intel FPGA IP* — *Implementing the GTS Reset Sequencer Intel FPGA IP* — *GTS PMA/FEC Direct PHY Intel FPGA IP Example Design*
2023.03.10	23.1	Initial release.