F-tile Architecture and PMA and FEC Direct PHY IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **23.2**

IP Version: **4.5.0**

Online Version
Send Feedback

UG-20315

683872
2023.06.26

Contents

1. F-tile Overview

This user guide describes architecture and implementation details for the Intel Agilex® 7 F-tile building blocks, physical (PHY) layer IP, PLLs, and clock networks. F-tile has up to 20 PMAs per tile, each with integrated advanced high-speed analog signal conditioning and clock data recovery circuits for chip-to-chip, chip-to-module, and backplane applications.

F-tile is a PAM4 and NRZ dual-mode serial interface tile that contains 16 FGT PMAs and four FHT PMAs. F-tile contains multiple hard IP blocks for use in conjunction with the PMAs to allow efficient implementation of popular and emerging serial protocols. F-tile connects to the FPGA fabric using the Intel embedded multi-die interconnect bridge (EMIB) technology.

Table 1. F-Tile Features

Feature	Description
Number of available PMAs	Up to 20. • FHT: up to four per tile. • FGT: up to 16 per tile. Not all FHT PMAs bond out in every tile. Refer to *Intel® Agilex™ 7 Device Family Pin Connection Guidelines*.
Data rate range	FHT: • 24-29 Gbps NRZ • 48-58 Gbps NRZ and PAM4 • 96-116 Gbps PAM4 FGT: • 1-32 Gbps NRZ • 20-58.125 PAM4 Not all FGT PMAs support the same data rates. Refer to *PMA Data Rates*.
Number of EMIBs	24
PCIe* hard IP modes	Up to one Gen4 x16, two Gen4 x8, or four Gen4 x4.
Ethernet hard IP modes with number of supported PMAs for each, where *10GbE-1* is 10GbE mode supporting one PMA	10GbE-1, 25GbE-1, 40GbE-4, 50GbE-2, 50GbE-1, 100GbE-4, 100GbE-2, 100GbE-1, 200GbE-8, 200GbE-4, 200GbE-2, 400GbE-8, and 400GbE-4, with these optional features: • auto-negotiation • link training • IEEE 1588 precision time protocol (PTP) Includes Ethernet PCS and MAC for all data rates. Not all features are supported for all data rates. Refer to *F-Tile Ethernet Intel FPGA Hard IP User Guide*.
Forward error correction (FEC) and Reed-Solomon FEC (RS-FEC) modes	• IEEE 802.3 BASE-R Firecode (CL 74) • Ethernet Technology Consortium (ETC) RS(272, 258) • IEEE 802.3 RS(528, 514) (CL91) • IEEE 802.3 RS(544, 514) (CL 134) Refer to *F-Tile Supported FEC Modes and Compliance Specifications*.

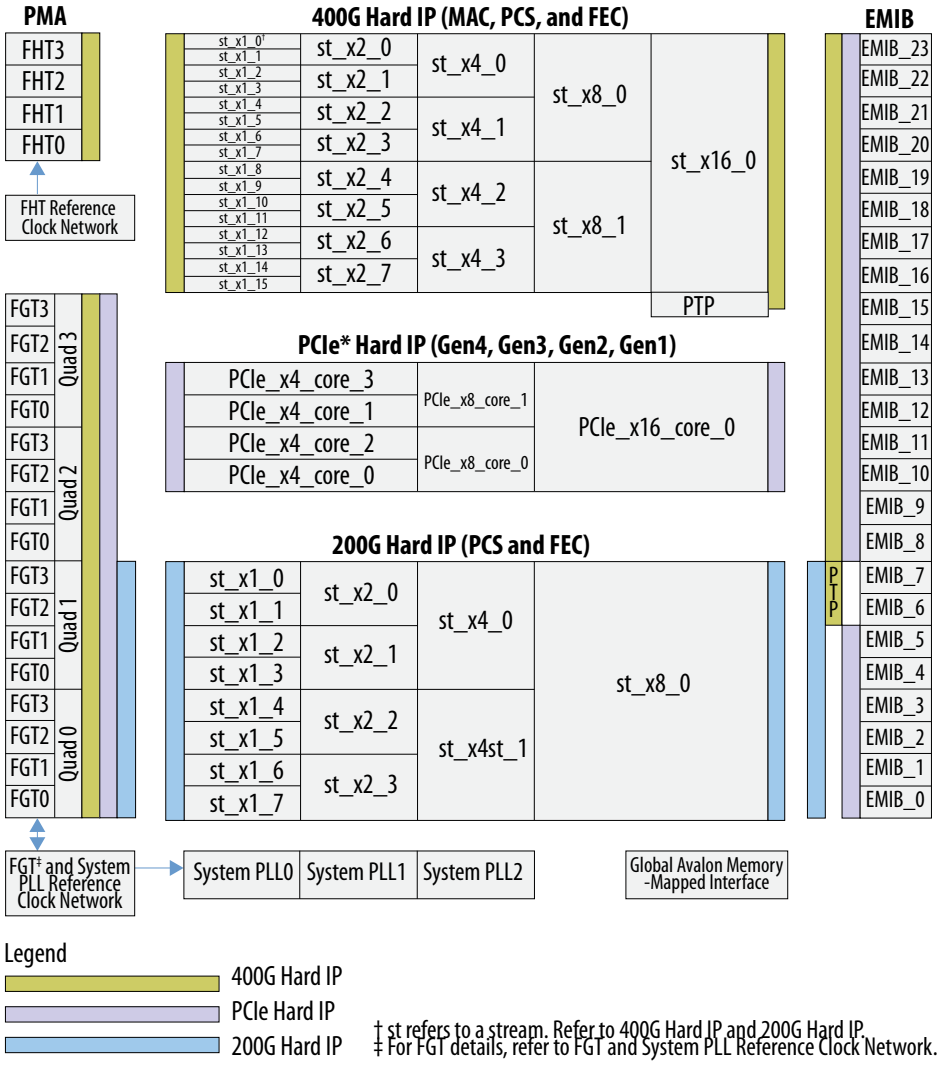
Related Information

- Intel Agilex 7 Device Data Sheet
- Intel Agilex 7 Device Family Pin Connection Guidelines
- Intel Agilex 7 Device Overview

2. F-tile Architecture

2.1. F-Tile Building Blocks

Figure 1. F-Tile Architecture Building Blocks



F-tile architecture building blocks include:

- PMAs
  - FGT
  - FHT
- Hard IPs
  - 400G hard IP
  - 200G hard IP
  - PCIe hard IP
- EMIB
- IEEE 1588 precision time protocol for Ethernet
- Clock networks
  - Reference clock network
  - Datapath clock network

Related Information

- 400G Hard IP and 200G Hard IP on page 10
- FGT and System PLL Reference Clock Network on page 69

2.1.1. FHT and FGT PMAs

There are up to four FHT PMAs per F-tile and up to 16 FGT PMAs per F-tile. Four contiguous FGT PMAs can be grouped together to form a quad. See *PMA Architecture* for details.

See *PMA Data Rates* for supported data rates.

Table 2. PMA Quads and Names

PMA	Quad	PMAs per Tile	PMA Names
FHT	N/A	up to 4	**FHT3**, **FHT2**, **FHT1**, **FHT0**
FGT	Quad3	up to 16	**FGT3_Quad3**, **FGT2_Quad3**, **FGT1_Quad3**, **FGT0_Quad3**
FGT	Quad2	up to 16	**FGT3_Quad2**, **FGT2_Quad2**, **FGT1_Quad2**, **FGT0_Quad2**
FGT	Quad1	up to 16	**FGT3_Quad1**, **FGT2_Quad1**, **FGT1_Quad1**, **FGT0_Quad1**
FGT	Quad0	up to 16	**FGT3_Quad0**, **FGT2_Quad0**, **FGT1_Quad0**, **FGT0_Quad0**

Related Information

- PMA Data Rates on page 14
- PMA Architecture on page 55

2.1.2. 400G Hard IP and 200G Hard IP

F-tile supports 400G hard IP and 200G hard IP. 400G hard IP interfaces either with up to four FHT PMAs or 16 FGT PMAs (one at a time), and 200G hard IP interfaces with up to eight FGT PMAs (Quad0 and Quad1).

Starting with Intel Quartus® Prime Pro Edition software version 23.1, F-tile supports the mixed transceiver mode in which the 400G Hard IP interfaces with up to four FHT PMAs and four FGT PMAs (Quad2).

400G hard IP and 200G hard IP can be divided into fractures. The building block of a fracture is a stream.

- A stream is a logical representation of a datapath connecting one EMIB to one PMA, so one stream maps to one EMIB and has a maximum of 32 Gbps data. One PMA can connect up to four streams.
- The PMA width determines the number of streams used. Each PMA with a width of 32 or less uses one stream. Each PMA with a width of 64 uses two streams. Each PMA with a width of 128 uses four streams. For example:
  - 10GbE needs a single stream.
  - 40GbE running on four 10 Gbps PMAs needs four streams.
  - 53 Gbps PAM4 requires a 64 PMA width, so it needs two streams.
  - 100 Gbps PAM4 requires a 128 PMA width, so it needs four streams.
- See the following "Hard IP Fractures and Streams" figures.
  - In 400G hard IP, **Stream0**-**Stream15** map to **EMIB_23**-**EMIB_8**, respectively.
  - In 200G hard IP, **Stream0**-**Stream7** map to **EMIB_7**-**EMIB_0**, respectively.

One or more streams combine to form a fracture. For example, 16 streams are combined to form the **st_x16** fracture. A fracture can include MAC, PCS, FEC, a combination of the three, or bypass all three. The fracture naming convention is:

```
<st as an abbreviation of stream>_x<number of streams in the fracture>_<index number>
```

For example, the **st_x2_2** fracture combines two streams of up to 32 Gbps data for index number two.

Each fracture in 400G hard IP includes MAC, PCS, and FEC. Each fracture in 200G hard IP includes a PCS and FEC. 400G hard IP is a superset of 200G hard IP. If you need MAC or require bandwidth greater than 200 Gbps, use 400G hard IP. See *PMA Data Rates* for supported data rates.

Table 3. Supported Fractures in 400G Hard IP and 200G Hard IP

Fracture Type	400G Hard IP (MAC, PCS, and FEC)		200G Hard IP (PCS and FEC)		Example Protocols
	Number of Fractures (*n*)	Number of Streams and EMIBs per Fracture	Number of Fractures (*n*)	Number of Streams and EMIBs per Fracture	
st_x1	16	1	8	1	9.8 Gbps PMA Direct 10GbE-1 IEEE Ethernet 25GbE-1 IEEE Ethernet 16G Fibre Channel 32G Fibre Channel
st_x2	8	2	4	2	50GbE-1 and 50GbE-2 IEEE Ethernet 50GbE-1 and 50GbE-2 Ethernet Technology Consortium 64G Fibre Channel
st_x4	4	4	2	4	40GbE-4 IEEE Ethernet 100GbE-1, 100GbE-2, and 100GbE-4 IEEE Ethernet 100GbE-2 Ethernet Technology Consortium 100G Interlaken with RS-FEC 128G Fibre Channel
st_x8	2	8	1	8	200GbE-2, 200GbE-4, and 200GbE-8 IEEE Ethernet 200GbE-4 Ethernet Technology Consortium
st_x16	1	16	N/A	N/A	400GbE-4 IEEE Ethernet 400GbE-8 Ethernet Technology Consortium 400G FlexO

Table 4. 400G Hard IP and 200G Hard IP Supported Configurations

Configuration	MAC (400G hard IP only)	PCS (no 8b/10b)	FEC	PMA	Example Protocols
Supported by 400G Hard IP					
Full hard IP (with MAC)	Yes	Yes	Optional	FHT or FGT	Ethernet
Supported by 400G Hard IP and 200G Hard IP					
PCS Direct	Not used	Yes	Optional	FHT or FGT for 400G Hard IP FGT for 200G Hard IP	Common Public Radio Interface (CPRI) 24G
FlexE	Not used	Yes (64b/66b encoder/decoder bypass)	Optional	FHT or FGT for 400G Hard IP FGT for 200G Hard IP	FlexE
Open Transport Network (OTN)	Not used	Yes (64b/66b encoder/decoder and scrambler bypass)	Optional	FHT or FGT for 400G Hard IP FGT for 200G Hard IP	OTN
FEC Direct	Not used	Not used	Yes	FHT or FGT for 400G Hard IP FGT for 200G Hard IP	Interlaken 300G-6, 32G Fibre Channel, 64G Fibre Channel
PMA Direct	Not used	Not used	Not used	FHT or FGT for 400G Hard IP FGT for 200G Hard IP	CPRI (low data rate), high-definition multimedia interface (HDMI), serial digital interface (SDI), JESD204B/C

Figure 2. 400G Hard IP Fractures and Streams

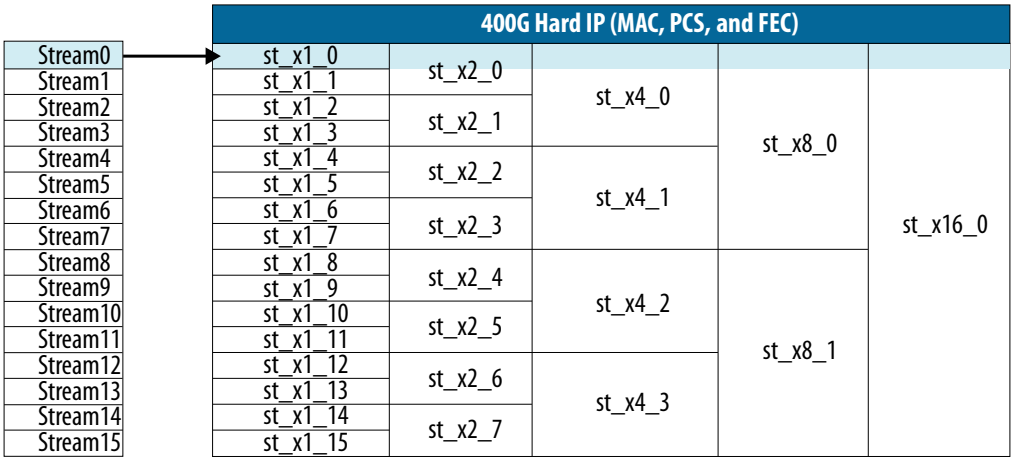
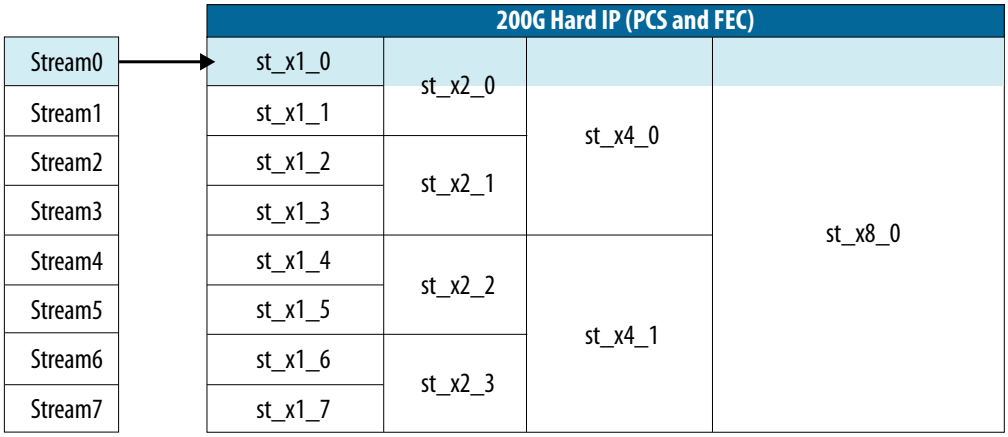


Figure 3. 200G Hard IP Fractures and Streams

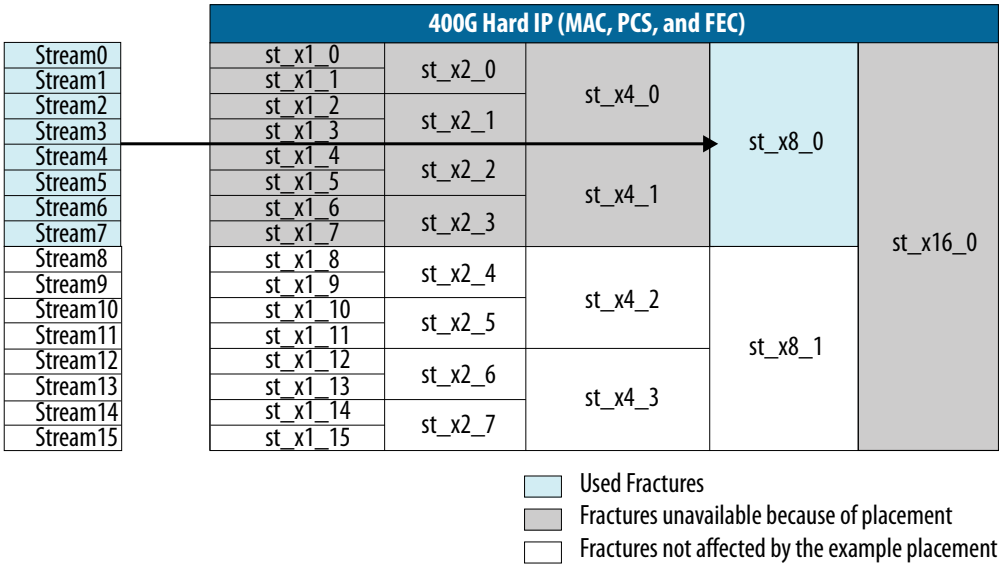


400G hard IP and 200G hard IP can use various fracture combinations. Using a fracture makes the fractures to its right and left that share the same stream unavailable. For example, for 400G hard IP, the **st_x8_0** and **st_x16_0** fractures share eight streams (**Stream0**-**Stream7**), and the **st_x2_0** and **st_x8_0** fractures share two streams (**Stream0**-**Stream1**).

Therefore, using **st_x8_0** in 400G hard IP makes the following fractures unavailable:

- **st_x16_0**
- **st_x4_0**
- **st_x4_1**
- **st_x2_0**-**st_x2_3**
- **st_x1_0**-**st_x1_7**

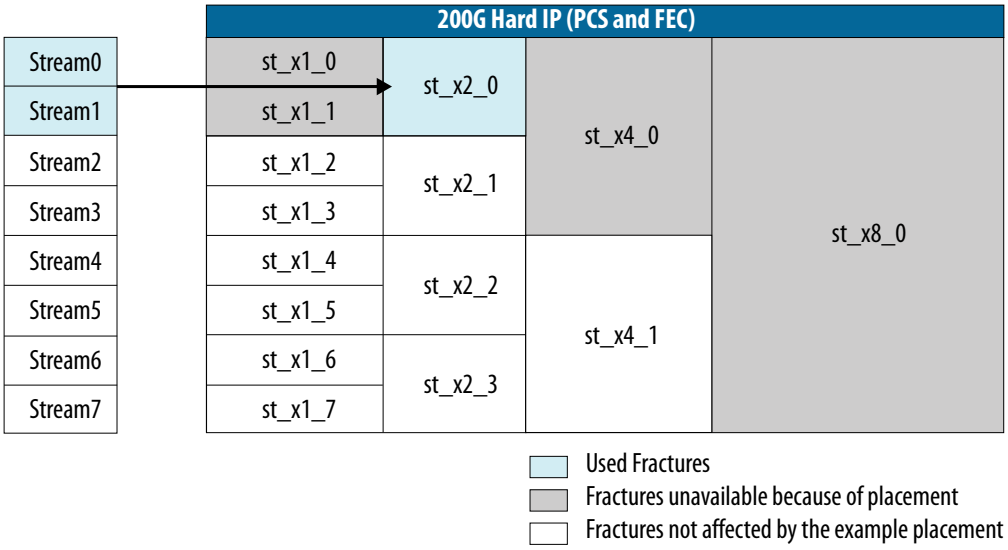
Figure 4. 400G Hard IP Usable Fractures by Stream



And, using **st_x2_0** in 200G hard IP makes the following fractures unavailable:

- **st_x8_0**
- **st_x4_0**
- **st_x1_1**
- **st_x1_0**

Figure 5. 200G Hard IP Usable Fractures by Stream



Related Information

PMA Data Rates on page 14

2.1.3. PMA Data Rates

The PMA data rate is dictated by the datapath clocking mode and PMA type you select.

For speed grade information, see the *Intel Agilex 7 Device Data Sheet*.

Table 5. PMA Data Rate Ranges by Datapath Clocking Mode

Datapath Clocking Mode	Configuration	Data Rate Range	
		FHT PMA	FGT PMA
PMA clocking mode (maximum 906.25 MHz)	PMA Direct	• 24-29 Gbps NRZ • 48-58 Gbps NRZ and PAM4 • 96-116 Gbps PAM4	• 1-29 Gbps NRZ(1) • 20-58 Gbps PAM4(2)
System PLL clocking mode (maximum 1 GHz)	PMA Direct Other configurations with MAC, PCS, and FEC		• 1-32 Gbps NRZ(1) • 20-58.125 PAM4(2)

(1) You must use the system PLL clocking mode for data rates between 29 Gbps and 32 Gbps NRZ.

(2) FGT **Quad0** can only support 20-32 Gbps PAM4. FGT **Quad1**, **Quad2**, and **Quad3** can support 20-58 Gbps PAM4.

Table 6. Fracture Type Used by Mode

Fracture Type	Ethernet Mode		PMA Direct Mode[3]		FEC Direct Mode	
	FGT	FHT	FGT	FHT	FGT	FHT
st_x1	10GbE-1, 25GbE-1	25GbE-1	One lane with PMA width ≤ 32	One lane with PMA width ≤ 32	One lane with PMA width ≤ 32	One lane with PMA width ≤ 32
st_x2	50GbE-1, 50GbE-2	50GbE-1, 50GbE-2	One lane with PMA width = 64	One lane with PMA width = 64	Two bonded lanes with PMA width ≤ 32 One lane with PMA width = 64	Two bonded lanes with PMA width ≤ 32 One lane with PMA width = 64
st_x4	40GbE-4, 100GbE-2, 100GbE-4	100GbE-1, 100GbE-2, 100GbE-4	Not used	One lane with PMA width = 128	Four bonded lanes with PMA width ≤ 32 Two bonded lanes with PMA width = 64	Four bonded lanes with PMA width ≤ 32 Two bonded lanes with PMA width = 64 One lane with PMA width = 128
st_x8	200GbE-4, 200GbE-8	200GbE-2, 200GbE-4	Not used	Not used	Eight bonded lanes with PMA width ≤ 32 Four bonded lanes with PMA width = 64	Four bonded lanes with PMA width = 64 Two bonded lanes with PMA width = 128
st_x16	200GbE-4, 400GbE-8	200GbE-4, 400GbE-4	Not used	Not used	Eight bonded lanes with PMA width = 64	Four bonded lanes with PMA width = 128

(3) When using a bonded configuration in PMA Direct mode, the number of bonded PMA lanes determines the number of fractures used. For example:

- Two, four, eight, and 16 bonded FGT lanes with a PMA width ≤ 32 use two, four, eight, and 16 **st_x1** fractures, respectively.
- Two, four, and eight bonded FGT lanes with a PMA width = 64 use two, four, and eight **st_x2** fractures, respectively.
- Two and four bonded FHT lanes with a PMA width ≤ 32 use two and four **st_x1** fractures, respectively.
- Two and four bonded FHT lanes with a PMA width = 64 use two and four **st_x2** fractures, respectively.
- Two and four bonded FHT lanes with a PMA width = 128 use two and four **st_x4** fractures, respectively.

PMA Direct modes, unlike Ethernet and FEC Direct modes, do not use the underlying logic of the fracture (MAC, PCS, or FEC). The following examples illustrate the differences between Ethernet and FEC Direct modes and PMA Direct modes.

- PMA width = 32 example:
  - 100GbE-4 Ethernet link using four PMAs – four bonded lanes of 25.78125 Gbps
    - One **st_x4** fracture is used.
  - 100 Gbps PMA Direct link using four PMAs – four bonded lanes of 25.78125 Gbps
    - Four **st_x1** fractures are used.
  - 100 Gbps FEC Direct link using four PMAs – four bonded lanes of 25.78125 Gbps
    - One **st_x4** fracture is used.
- PMA width = 64 example
  - 400GbE-8 Ethernet link using eight PMAs – eight bonded lanes of 53.125 Gbps
    - One **st_x16** fracture is used.
  - 400 Gbps PMA Direct link using eight PMAs – eight bonded lanes of 53.125 Gbps
    - Eight **st_x2** fractures are used.
  - 400 Gbps FEC Direct link using eight PMAs – eight bonded lanes of 53.125 Gbps
    - One **st_x16** fracture is used.

2.1.4. FEC Architecture

Each 400G hard IP and 200G hard IP fracture includes FEC. The FEC block is located between the PCS and PMA interface blocks of a fracture.

- Four adjacent **st_x1** fractures or two adjacent **st_x2** fractures share the same clock and are grouped together as a FEC core.
- **st_x4** uses one FEC core.
- Two or four FEC cores are grouped together in **st_x8** and **st_x16** fractures, respectively.

There is a total of four FEC cores in 400G hard IP and two FEC cores in 200G hard IP. Each FEC core can be used to implement multiple FEC modes as shown in the following table.

Table 7. F-Tile Supported FEC Modes and Compliance Specifications

FEC Mode	Specification	FEC Compliance Specification	Example Protocols
Firecode	IEEE	IEEE 802.3 BASE-R Firecode (CL 74)	25GbE-1 IEEE BASE-R
RS(272, 258) LL	ETC	ETC RS(272,258)	50GbE-1, 100GbE-2, 200GbE-4, 400GbE-8 ETC
RS(528, 514) KR	IEEE	IEEE 802.3 RS(528, 514) (CL 91)	25GbE-1
	ETC	IEEE 802.3 RS(528, 514) (CL 91) ETC	25GbE-1, 50GbE-2 ETC

continued...

FEC Mode	Specification	FEC Compliance Specification	Example Protocols
	Fibre Channel	Fibre Channel RS(528, 514)	Fibre Channel 16G, 32G, 64G and 128G CPRI 10.1376 Gbps and 24.33024 Gbps
	FlexO	FlexO RS(528, 514)	OTU25 100G FlexO
RS(544, 514) KP	IEEE	IEEE 802.3 RS(544,514) (CL 134)	50GbE-1 100GbE-1, 100GbE-2 200GbE-4 400GbE-4, 400GbE-8 OTU25u
	Custom	Custom IEEE 802.3 RS(544, 514) (CL 134) at 26.5625 Gbps NRZ	25GbE-1 50GbE-2 100GbE-4 200GbE-8
	Interlaken	Interlaken RS(544, 514)	Interlaken (100G bundles)
	Fibre Channel	Fibre Channel RS(544, 514)	16G, 32G, and 128G Fibre Channel
	FlexO	FlexO RS(544, 514)	100G FlexO (4x 25G NRZ, 2x 50G PAM4, 1x 100G PAM4)

Note: The table above has a few examples of the list of protocols supported by F-tile. The list is not an exhaustive list of protocols supported.

If your configuration has multiple interface in one FEC core, you need custom cadence. Refer to *Datapath Clock Cadences* for details. Refer to *FEC Placement Rules* for examples.

Related Information

- Datapath Clock Cadences on page 74
- FEC Placement Rules on page 47

2.1.5. PCIe Hard IP

The F-tile PCIe hard IP consists of four PCIe cores: one x16 (**core_0**), one x8 (**core_1**) and two x4 (**core_2**, **core_3**). It consists of a set of port bifurcation muxes to remap the four controller PHY interface for PCI Express* (PIPE) lane interfaces to the shared 16 FGT lanes. **core_0** can be configured to support x16, x8, and x4 configurations, and **core_1** can be configured to support x8 and x4 configurations. **core_2** and **core_3** only support only x4 configurations.

Figure 6. PCIe Hard IP (Gen4, Gen3, Gen2, and Gen1) Configurations

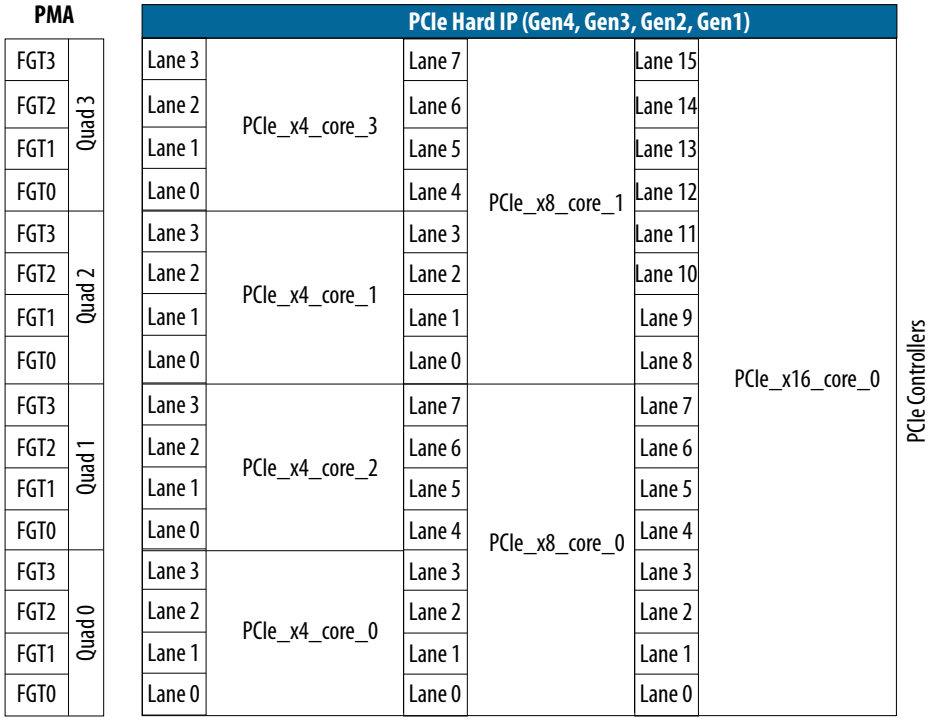


Table 8. PCIe Hard IP (Gen4, Gen3, Gen2, and Gen1) Supported Configurations

Configuration	Interface Type
1x PCIe x16	Root port or endpoint and upstream or downstream port
2x PCIe x8	Endpoint only Upstream/upstream port Downstream/downstream port Endpoint/upstream port Upstream/downstream port
1x PCIe x8	Root port or endpoint
4x PCIe x4	Root port only and upstream or downstream port
2x PCIe x4	Root port only
1x PCIe x4	Endpoint only

Table 9. PCIe Hard IP Layers by Mode

Mode	Transaction Layer	Data Link Layer	PHY Layer
Full hard IP	Yes	Yes	Yes
Transaction layer packet (TLP) bypass	Yes (Lite)	Yes	Yes

2.1.6. Bonding Architecture

Streams or PMAs are bonded together under the following conditions:

- When a transmitter (TX) or receiver (RX) PMA lane operates at a data rate above 32 Gbps in the high data rate PAM4 mode (PMA bonding)
- When there are multiple TX PMA lanes in PAM4 or NRZ mode (system bonding)

For PMA bonding when a TX or RX PMA lane operates at a data rate above 32 Gbps, use:

- Two streams for up to 58 Gbps (in FGT and FHT PMAs)
- Four streams for up to 116 Gbps (in FHT PMAs only)

System bonding for multiple TX PMA lanes is a technique used to minimize high-speed, serial, lane-to-lane transmit skew for multi-lane protocols (see the *Intel Agilex 7 Device Data Sheet* for the skew specification). This technique bonds multiple streams from multiple TX PMA lanes together. For example:

- In 200G hard IP Ethernet with four FGT PMA lanes where each PMA lane has two streams, eight streams are bonded together.
- In 400G hard IP Ethernet with four FHT PMA lanes where each PMA lane has four streams, 16 streams are bonded together.

Any reference clock on the same tile can be used for the bonded lanes provided that its clock network spans all bonded lanes. Refer to *Clock Networks* for information about the reference clock network span.

When bonded streams use a system PLL, they must all use the same system PLL. For example, six-stream, JESD204B, TX-only IP uses the same system PLL for all six TX streams.

Whenever bonding, all things in the bond must be on the same clock.

You can only bond lanes within the same tile, either within the 16 FGT PMA lanes or within the four FHT PMA lanes. TX PMA bonded lanes must be placed contiguously with the primary PMA lane at a pre-defined location based on the fracture type. See the *F-Tile Channel Placement Tool* and *Bonding Placement Rules* for details.

Unsupported bonding conditions:

- Bonding between FGT and FHT PMA lanes
- Bonding between 400G hard IP and 200G hard IP

There are three mechanisms to facilitate bonding:

- Sharing a TX and RX PMA recovered clock network at the PMA interface
- Synchronizing a reset or read and write at the PMA interface
- Removing skew with EMIB interface deskew logic

400G hard IP, 200G hard IP, and PCIe hard IP all have TX and RX deskew logic. The deskew logic removes the skew of multiple bonded streams across the EMIB.

Table 10. Number of Bonded Streams Supported by Deskew Logic in 400G Hard IP and 200G Hard IP

Hard IP	Number of Supported Bonded Streams in 400G Hard IP	Number of Supported Bonded Streams in 200G Hard IP
F-tile PMA and FEC Direct PHY IP	2, 4, 6, 8, 12, 16	2, 4, 6, 8
F-tile Ethernet hard IP	2, 4, 8, 16	2, 4, 8

Related Information

- Clock Networks on page 22
- Bonding Placement Rules on page 49
- Intel Agilex 7 Device Data Sheet
- F-Tile Channel Placement Tool

2.1.7. Deskew Logic

When multiple streams are bonded together, the skew between multiple EMIB interfaces must be removed. The F-tile PMA Direct and FEC Direct IPs automatically compensate for the skew between multiple EMIB interfaces without the need for intervention.

TX Deskew Logic

Once bonding is enabled, deskew logic in the transmit direction aligns the EMIB interface by generating internal deskew pulses every 16 or 32 clock cycles (depending on variant configuration) for all TX EMIB interfaces. These pulses are used to align the transmit parallel data before insertion into the PMA. This logic is generated for all IP modes.

RX Deskew Logic

Once bonding is enabled, optional deskew logic in the receive direction aligns the EMIB interface based on deskew pulses received in the RX data. In PMA Direct mode, this optional RX deskew is implemented for PAM4 mode when the number of EMIB interfaces per PMA is greater than one. In PMA Direct mode, the deskew logic operates on EMIB interfaces associated with each PMA independently. For example, for a 2-lane PMA Direct variant, the deskew logic deskews the EMIB interfaces for Lane 1 independently from the EMIB interfaces for Lane 2. In FEC Direct mode, this optional RX deskew is implemented when the total number of EMIB interfaces is greater than one. In FEC Direct Mode, the RX deskew logic aligns all the EMIB interfaces together. For example, for a 2-lane FEC Direct variant, the deskew logic deskews the EMIB interfaces for Lane 1 together with the EMIB interfaces from Lane 2. Whenever RX deskew is enabled, all `rx_coreclkin` inputs are expected to be driven by the same source. The RX Deskew logic can deskew up to five `rx_coreclkin` cycles.

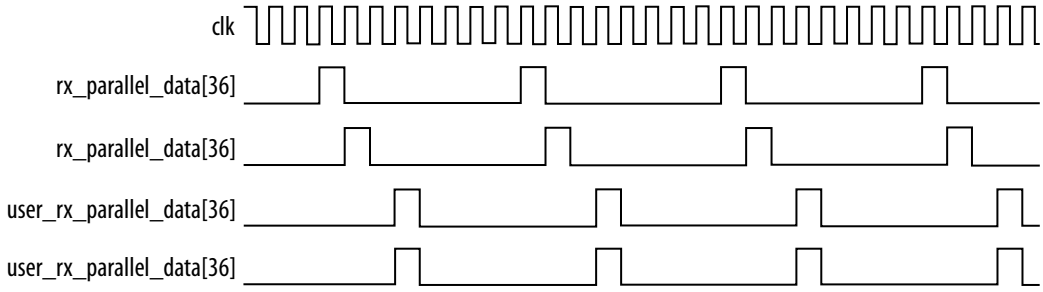
When operating in NRZ mode with no RS-FEC, deskew logic is always disabled.

In the RX direction, when operating in either PAM4 mode or FEC Direct Mode, deskew logic can be turned on with **Enable RX de-skew when available** parameter in the IP parameter editor. If the current configuration does not support RX de-skew, the parameter editor displays an information message during generating indicating that RX deskew is not supported for the current configuration.

When the deskew logic is disabled, you must deskew the `rx_parallel_data` buses coming out of each EMIB based on the RX deskew pulses provided by the IP. These deskew pulses appear on `rx_parallel_data[78]` when the PMA is in double data width mode and `rx_parallel_data[36]` when the PMA is in single width mode.

The following figure shows sample behavior on `rx_parallel_data` in single width mode before and after re-alignment.

Figure 7. Single Data Width Transfer



2.1.8. Embedded Multi-die Interconnect Bridge (EMIB)

An EMIB connects a stream in F-tile to the FPGA core. An F-tile has 24 EMIB streams. An EMIB stream can be mapped to one or more hard IPs. See the figure in *F-Tile Building Blocks* for EMIB-to-hard-IP mapping.

Related Information

F-Tile Building Blocks on page 8

2.1.9. IEEE 1588 Precision Time Protocol for Ethernet

F-tile supports the IEEE 1588 precision time protocol feature for Ethernet hard IP. This feature can only be used by 400G hard IP with MAC enabled. Enabling the PTP feature places a limit on how many hard IPs can be implemented using 200G hard IP because the PTP feature uses **EMIB_6** and **EMIB_7**.

Refer to *Topologies* for more details. Refer to *F-Tile Ethernet Intel FPGA Hard IP User Guide* for Ethernet modes supported with IEEE 1588 precision time protocol.

Related Information

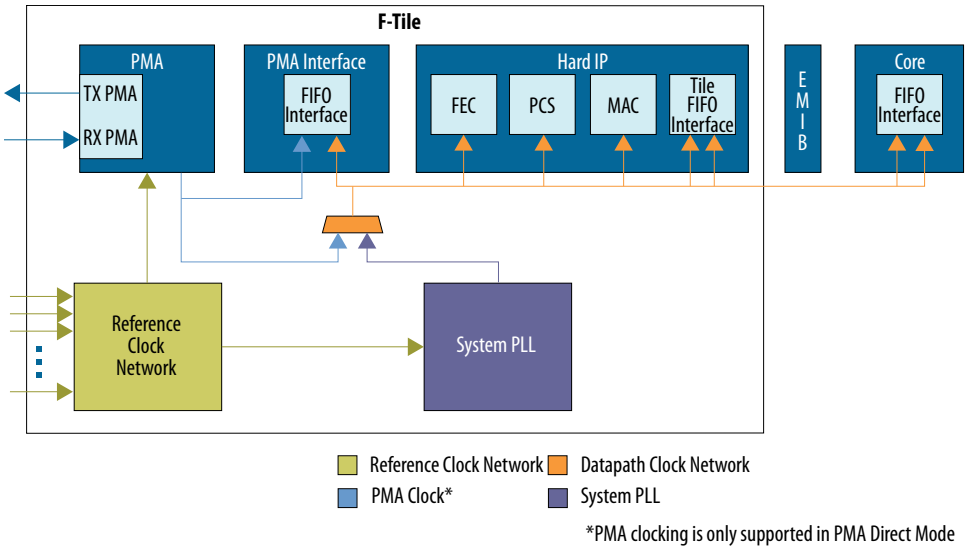
- Topologies on page 43
- F-Tile Ethernet Intel FPGA Hard IP User Guide

2.1.10. Clock Networks

F-tile has two types of clock network: reference and datapath.

- The reference clock network connects the physical reference clock pins from the device to the PMA. There are two types of reference clock networks:
  - FHT reference clock network for FHT PMAs
  - FGT and system PLL reference clock network for FGT PMAs and system PLLs
- The datapath clock network is accessible by all enabled digital blocks and is driven by a clock from either the PMA or one of three system PLLs.

Figure 8. Clock Network



Refer to *Clock Architecture* and *F-Tile Clocking Tool* for details.

Related Information

- Clock Architecture on page 67
- F-Tile Clocking Tool

2.1.11. Reconfiguration Interfaces

Each EMIB has one datapath Avalon® memory-mapped interface and one PMA Avalon memory-mapped interface, and each F-tile has one global Avalon memory-mapped interface.

- The datapath Avalon memory-mapped interface can access 400G hard IP (MAC, PCS, and FEC), 200G hard IP (PCS and FEC), EMIBs (both tile and core sides), PMA interfaces, and control and status registers (CSRs) implemented in the FPGA core.
- The PMA Avalon memory-mapped interface can access PCIe hard IP and PMAs.

Figure 9. F-Tile Architecture Building Blocks with Reconfiguration Interfaces

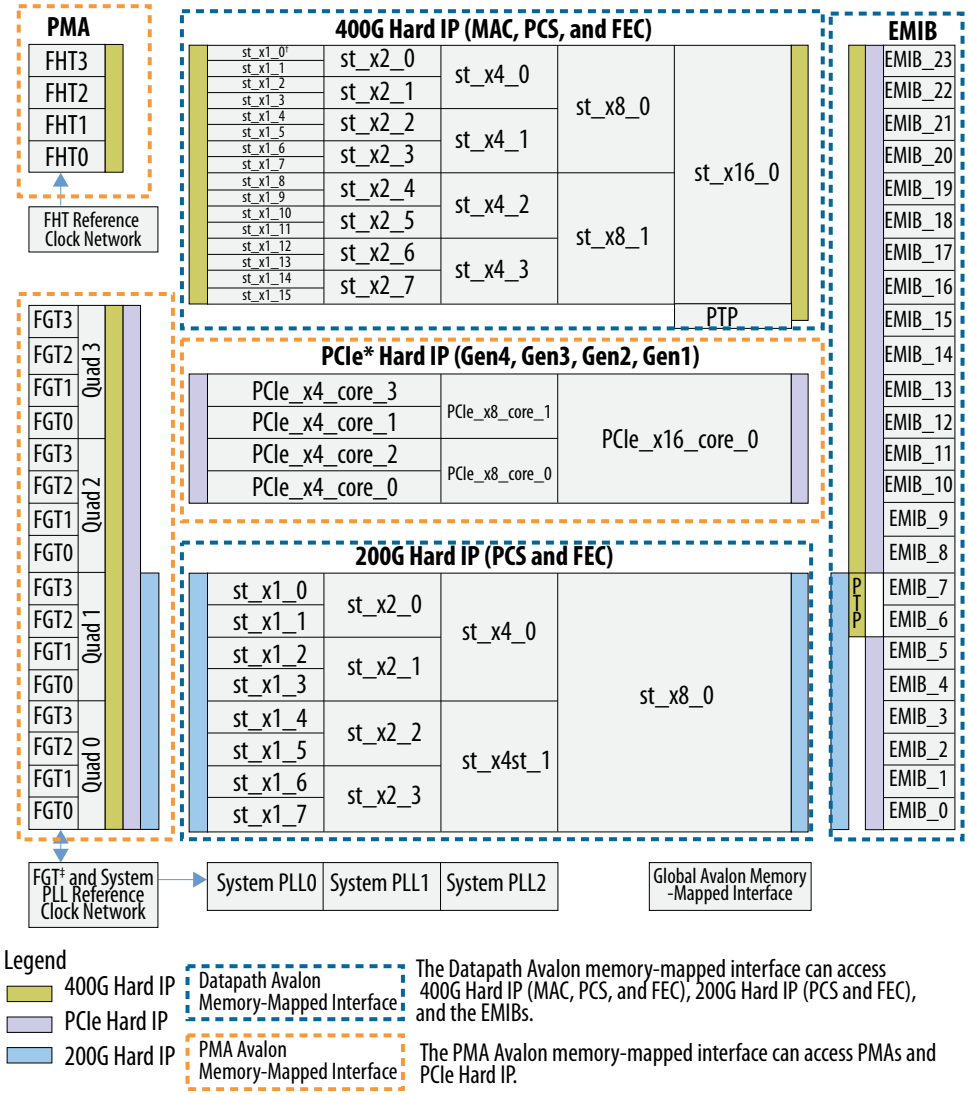
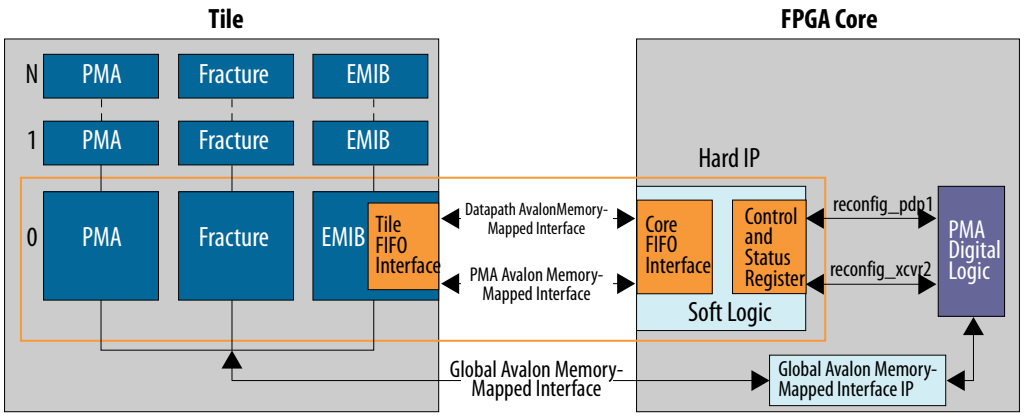


Figure 10. Tile-to-FPGA-Core Interfaces



Notes:
1. reconfig_pdp is connected to the Datapath Avalon Memory Mapped Interface.
2. reconfig_xcvr is connected to the PMA Avalon Memory Mapped Interface.

There is one `reconfig_pdp` per hard IP instance and one `reconfig_xcvr` per PMA. `reconfig_pdp` provides the parallel datapath interface of both 400G hard IP and 200G hard IP access to the datapath Avalon memory-mapped interface. `reconfig_pdp` is specific to Direct PHY IP. Other interfaces may use different naming. Ethernet, for example, uses `reconfig_eth`.

2.2. F-Tile Placement Rules

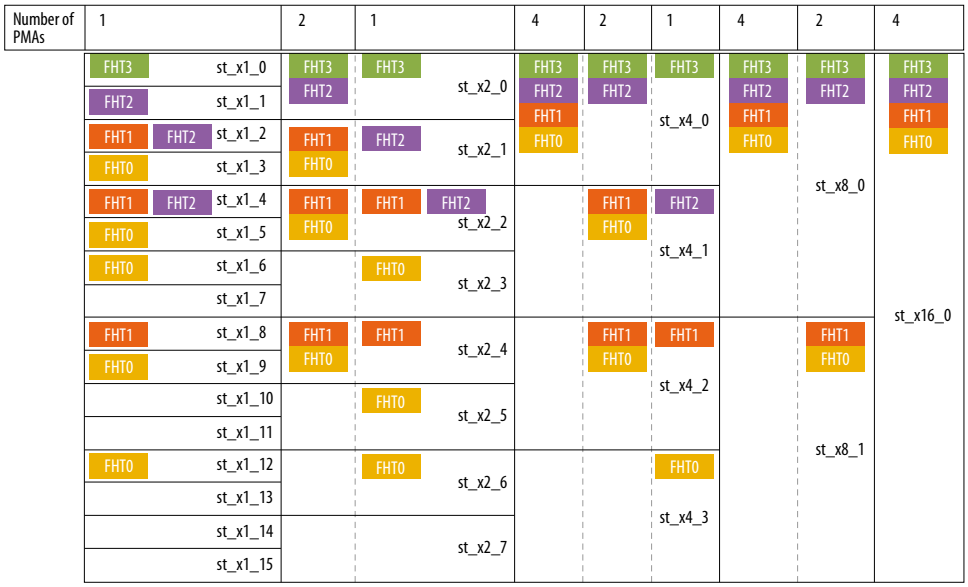
2.2.1. PMA-to-Fracture Mapping

PMA-to-fracture mapping defines how fractures can be mapped to PMAs. F-tile has flexibility for this mapping but there are restrictions.

- Not all PMAs map to all fractures
- Not all fractures map to all PMAs
- **st_x16** maps to a set of four FHT PMAs, eight FGT PMAs, or 16 FGT PMAs, or four FHT PMAs and four FGT PMAs (mixed transceiver mode).
- **st_x8** maps to a set of four FHT PMAs, two FHT PMAs, eight FGT PMAs, or four FGT PMAs.
- **st_x4** maps to a set of four FHT PMAs, two FHT PMAs, one FHT PMA, four FGT PMAs, or two FGT PMAs.
- **st_x2** maps to a set of two PMAs or one PMA.
- **st_x1** maps to one PMA.

2.2.1.1. FHT-PMA-to-400G-Hard-IP-Fracture Mapping

Figure 11. FHT-PMA-to-400G-Hard-IP-Fracture Mapping



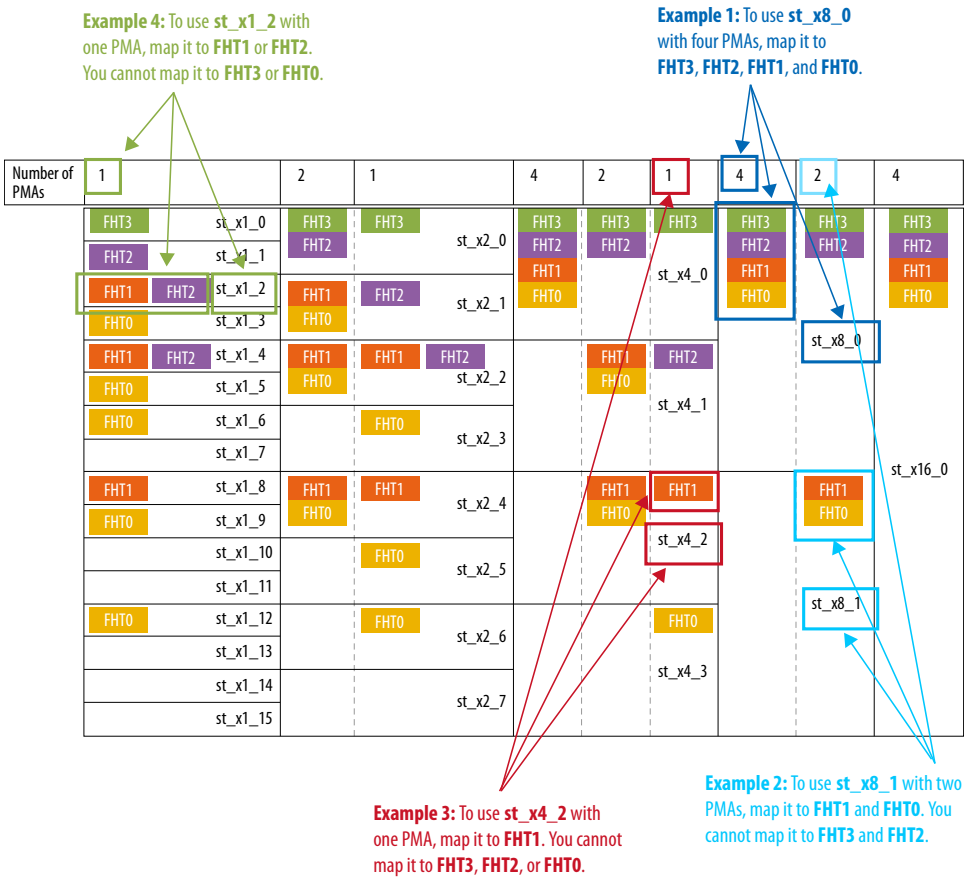
To read the previous figure and understand the PMA-to-fracture mapping options:

1. Select the fracture type and index you want.
2. Select the column for the number of PMAs you need.

The figure shows which PMAs are available to be mapped to the fracture and index you selected (and shows which PMAs are unavailable). For example:

- You want **st_x8_0**, and you need four PMAs. According to the figure, **FHT3**, **FHT2**, **FHT1**, and **FHT0** are available, so this is a possible mapping.
- However, you wanted **st_x8_1**, and you needed four PMAs, according to the figure, it is not a possible mapping because there is no set of four PMAs available for **st_x8_1**.

Figure 12. Understanding FHT-PMA-to-400G-Hard-IP-Fracture Mapping



FHT-PMA-to-400G-Hard-IP-Fracture Mapping Examples

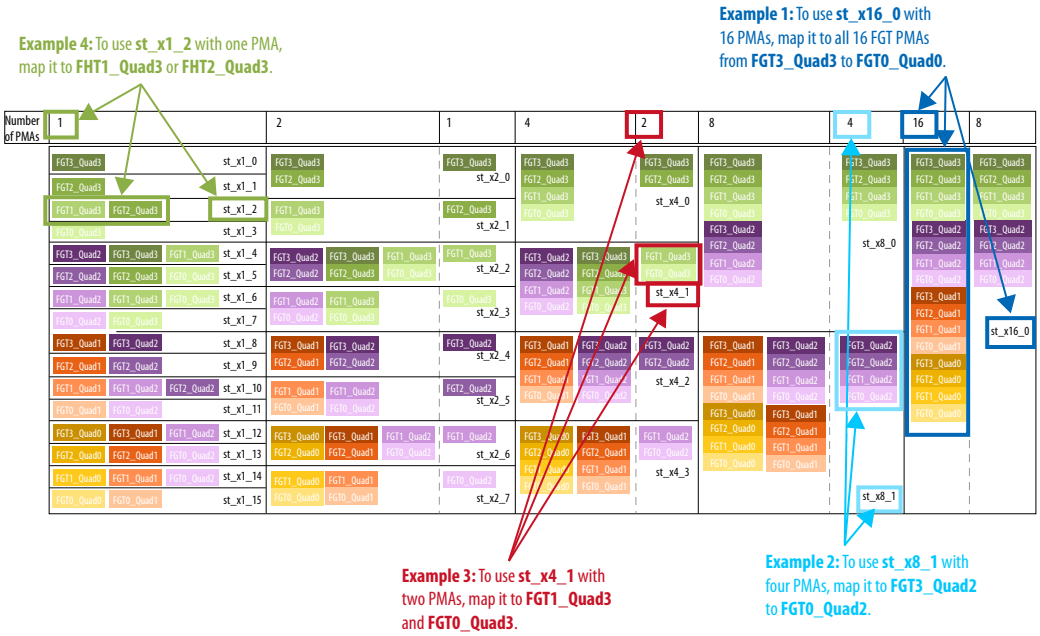
- **Example 1:** If you need the **st_x8_0** fracture with four PMAs, map it to **FHT3**, **FHT2**, **FHT1**, and **FHT0**.
- **Example 2:** If you need the **st_x8_1** fracture with two PMAs, map it to **FHT1** and **FHT0**. You cannot use **FHT3** and **FHT2** with **st_x8_1**.
- **Example 3:** If you need the **st_x4_2** fracture with one PMA, map it to **FHT1**. You cannot use **FHT3**, **FHT2**, or **FHT0**.
- **Example 4:** If you need the **st_x1_2** fracture with one PMA, map it to **FHT1** or **FHT2**. You cannot use **FHT3** or **FHT0**.

2.2.1.2. FGT-PMA-to-400G-Hard-IP-Fracture Mapping

Figure 13. FGT-PMA-to-400G-Hard-IP-Fracture Mapping

400G hard IP can be mapped to up to 16 FGT PMAs.

Figure 14. Understanding FGT-PMA-to-400G-Hard-IP-Fracture Mapping



FGT-PMA-to-400G-Hard-IP-Fracture Mapping Examples

- **Example 1:** If you need the **st_x16_0** fracture with 16 PMAs, map it to all 16 FGT PMAs from **FGT3_Quad3** through **FGT0_Quad0**.
- **Example 2:** If you need the **st_x8_1** fracture with four PMAs, map it to **FGT3_Quad2** through **FGT0_Quad2**.
- **Example 3:** If you need the **st_x4_1** fracture with two PMAs, map it to **FGT1_Quad3** and **FGT0_Quad3**.
- **Example 4:** If you need the **st_x1_2** fracture with one PMA, map it to **FGT1_Quad3** or **FGT2_Quad3**.

2.2.1.3. FGT-PMA-to-200G-Hard-IP-Fracture Mapping

Figure 15. FGT-PMA-to-200G-Hard-IP-Fracture Mapping

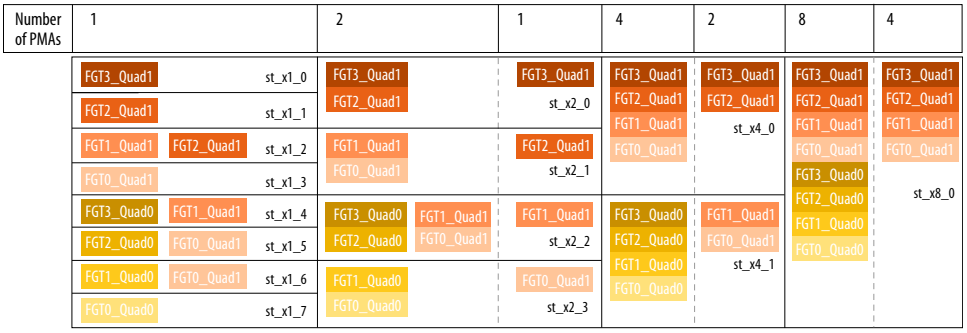
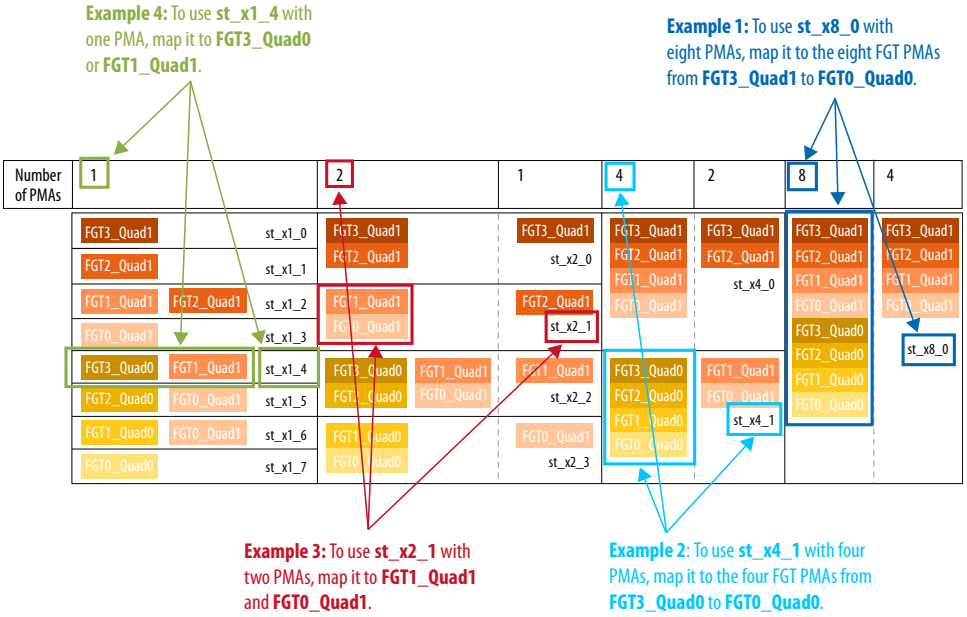


Figure 16. Understanding FGT-PMA-to-200G-Hard-IP-Fracture Mapping

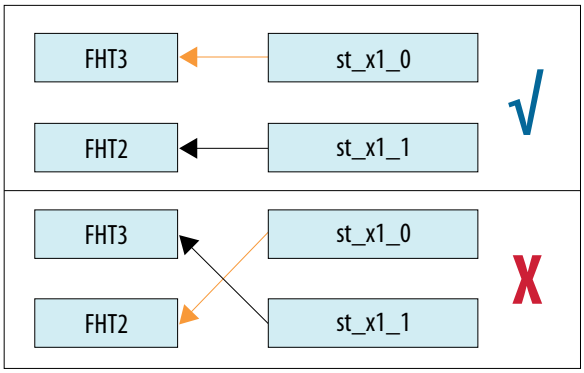


- **Example 1:** If you need the **st_x8_0** fracture with eight PMAs, map it to the eight FGT PMAs from **FGT3_Quad1** through **FGT0_Quad0**.
- **Example 2:** If you need the **st_x4_1** fracture with two PMAs, map it to **FGT1_Quad1** and **FGT0_Quad1**.
- **Example 3:** If you need the **st_x2_1** fracture with two PMAs, map it to **FGT1_Quad1** and **FGT0_Quad1**.
- **Example 4:** If you need the **st_x1_4** fracture with one PMA, map it to **FGT3_Quad0** or **FGT1_Quad1**.

2.2.2. Determining Which PMA to Map to Which Fracture

1. Determine the hard IP and PMA type you need (400G hard IP with FHT, 400G hard IP with FGT, or 200G hard IP with FGT).
2. Determine the number of PMAs you need to implement your interface.
3. Determine the fracture type (for example, **st_x16** or **st_x8**) using the "Fracture Type Used by Mode" table.
4. Identify possible fracture indices of the required fracture type that can map to the required number of PMAs. (The examples in the following sections explain this.)
5. If there is only one possible fracture index that meets your requirement, you must use that fracture index.
6. As you place each fracture, keep in mind that each placement blocks other fracture placements.
7. If there are multiple fracture indices that meet your requirements:
   a. Place your interfaces' fractures from top to bottom.
   b. Place your fractures in such a way as to be able to place all interfaces. Keep in mind that each placement blocks other fracture placements.
   c. Do not create crisscross connections when mapping fractures to PMAs.

Figure 17. PMA-to-Fracture Connections Example



   d. Place your higher data rate interfaces first.
   e. When placing interfaces with the same data rate (same fracture type), start placing interfaces that use a higher number of PMAs first. For example, place 100GbE-2 before 100GbE-1.

Related Information

PMA Data Rates on page 14
See the "Fracture Type Used by Mode" table.

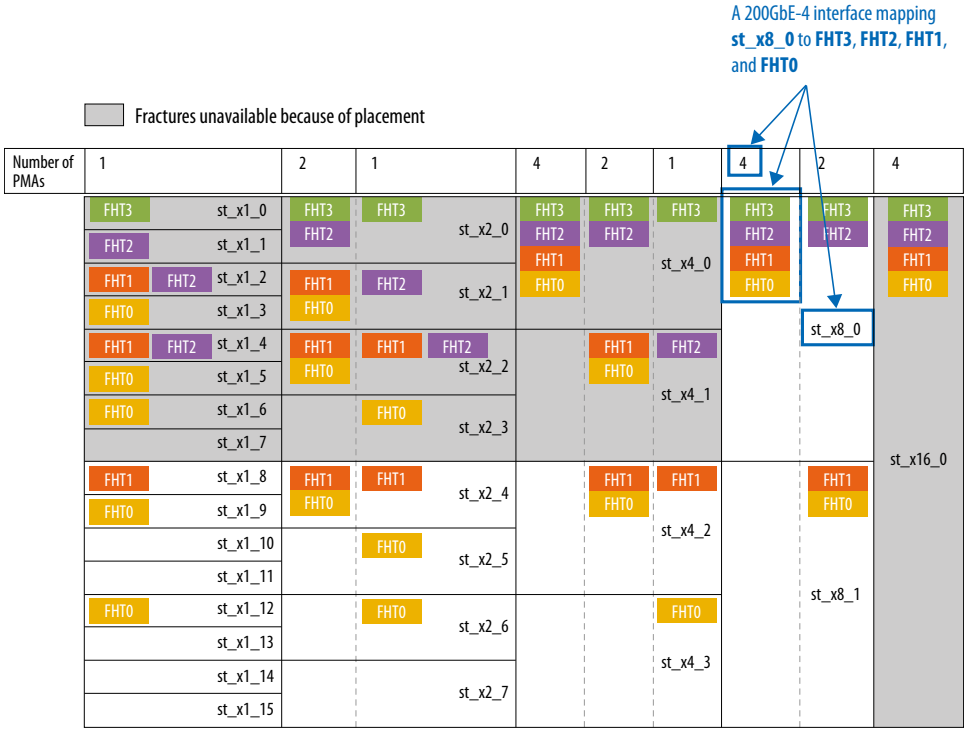
2.2.2.1. Implementing One 200GbE-4 Interface with 400G Hard IP and FHT

This example needs:

- Fracture type **st_x8**
- Four PMAs

Fracture Index	PMA Mapping
The only possible fracture index and PMA mapping:	
st_x8_0	**FHT3**, **FHT2**, **FHT1**, **FHT0**
The fracture index and PMA mapping that is not possible:	
st_x8_1	**FHT3**, **FHT2**, **FHT1**, **FHT0**

Figure 18. Implementing One 200GbE-4 Interface with 400G Hard IP and FHT



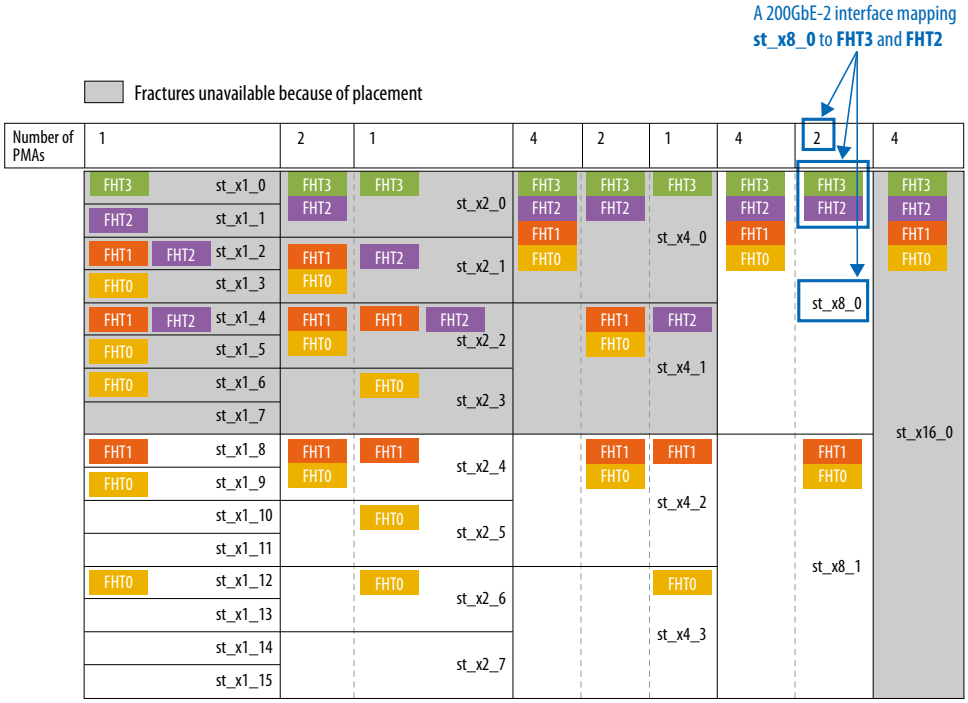
2.2.2.2. Implementing One 200GbE-2 Interface with 400G Hard IP and FHT

This example needs:

- Fracture type **st_x8**
- Two PMAs

Fracture Index	PMA Mapping
Only possible fracture indices and PMA mappings:	
st_x8_0	**FHT3**, **FHT2**
st_x8_1	**FHT1**, **FHT0**
Some fracture indices and PMA mappings that are not possible:	
st_x8_0	**FHT2**, **FHT1**
st_x8_0	**FHT1**, **FHT0**
st_x8_1	**FHT3**, **FHT2**

Figure 19. Implementing One 200GbE-2 Interface with 400G Hard IP and FHT



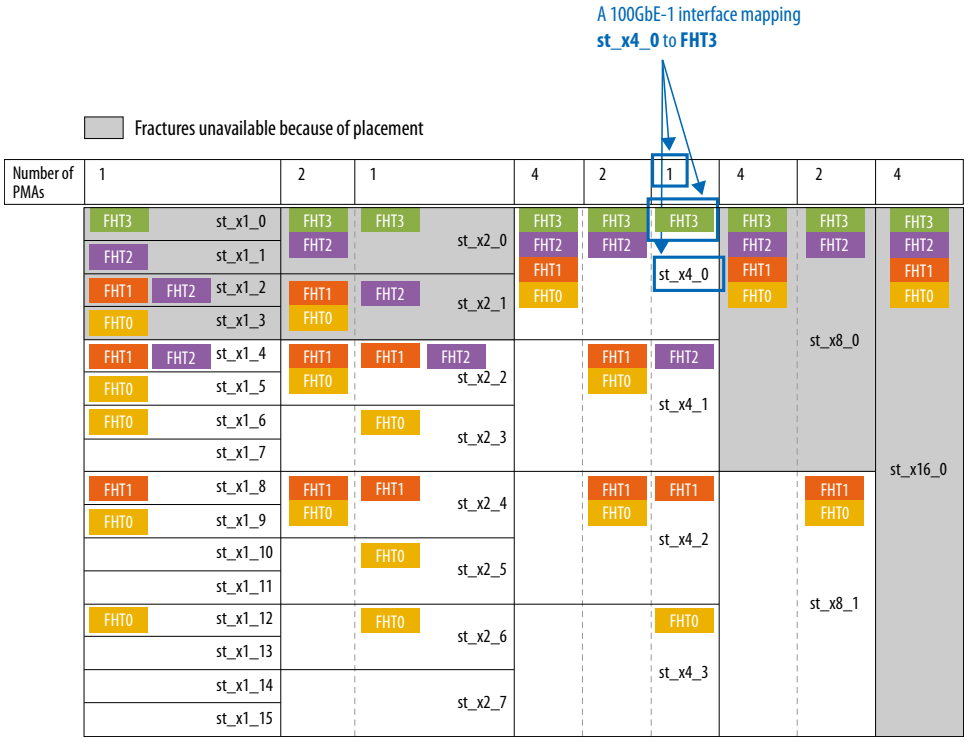
2.2.2.3. Implementing One 100GbE-1 Interface with 400G Hard IP and FHT

This example needs:

- Fracture type **st_x4**
- One PMA

Fracture Index	PMA Mapping
Some possible fracture indices and PMA mappings:	
st_x4_0	**FHT3**
st_x4_1	**FHT2**
Some fracture indices and PMA mappings that are not possible:	
st_x4_0	**FHT2**
st_x4_2	**FHT3**

Figure 20. Implementing One 100GbE-1 Interface with 400G Hard IP and FHT



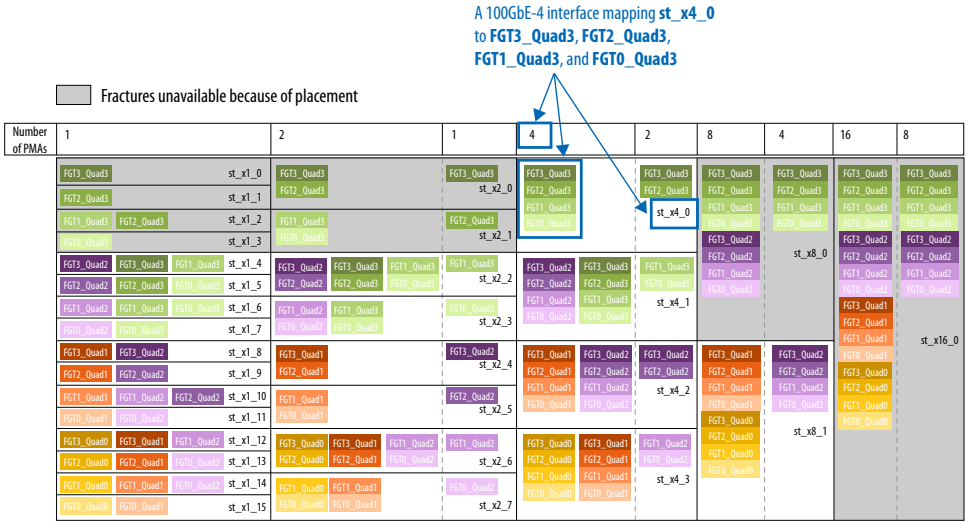
2.2.2.4. Implementing One 100GbE-4 Interface with 400G Hard IP and FGT

This example needs:

- Fracture type **st_x4**
- Four PMAs

Fracture Index	PMA Mapping
Some possible fracture indices and PMA mappings:	
st_x4_0	**FGT3_Quad3**, **FGT2_Quad3**, **FGT1_Quad3**, **FGT0_Quad3**
st_x4_1	**FGT3_Quad2**, **FGT2_Quad2**, **FGT1_Quad2**, **FGT0_Quad2**
st_x4_1	**FGT3_Quad3**, **FGT2_Quad3**, **FGT1_Quad3**, **FGT0_Quad3**
Some fracture indices and PMA mappings that are not possible:	
st_x4_0	**FGT3_Quad2**, **FGT2_Quad2**, **FGT1_Quad2**, **FGT0_Quad2**
st_x4_3	**FGT3_Quad2**, **FGT2_Quad2**, **FGT1_Quad2**, **FGT0_Quad2**

Figure 21. Implementing One 100GbE-4 Interface with 400G Hard IP and FGT



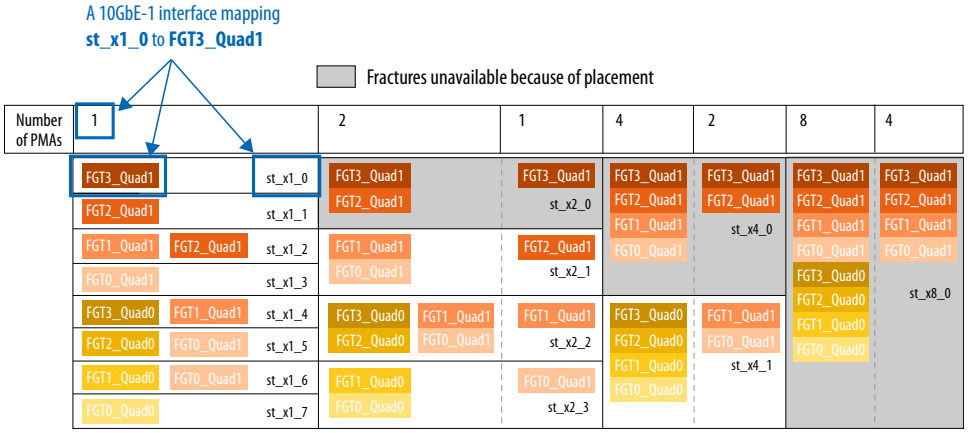
2.2.2.5. Implementing One 10GbE-1 Interface with 200G Hard IP and FGT

This example needs:

- Fracture type **st_x1**
- One PMA

Fracture Index	PMA Mapping
Some possible fracture indices and PMA mappings:	
st_x1_0	**FGT3_Quad1**
st_x1_2	**FGT1_Quad1**
st_x1_2	**FGT2_Quad1**
Some fracture indices and PMA mappings that are not possible:	
st_x1_0	**FGT2_Quad1**
st_x1_2	**FGT3_Quad1**

Figure 22. Implementing One 10GbE-1 Interface with 200G Hard IP and FGT



2.2.2.6. Implementing Three 25GbE-1 Interfaces with 400G Hard IP and FHT

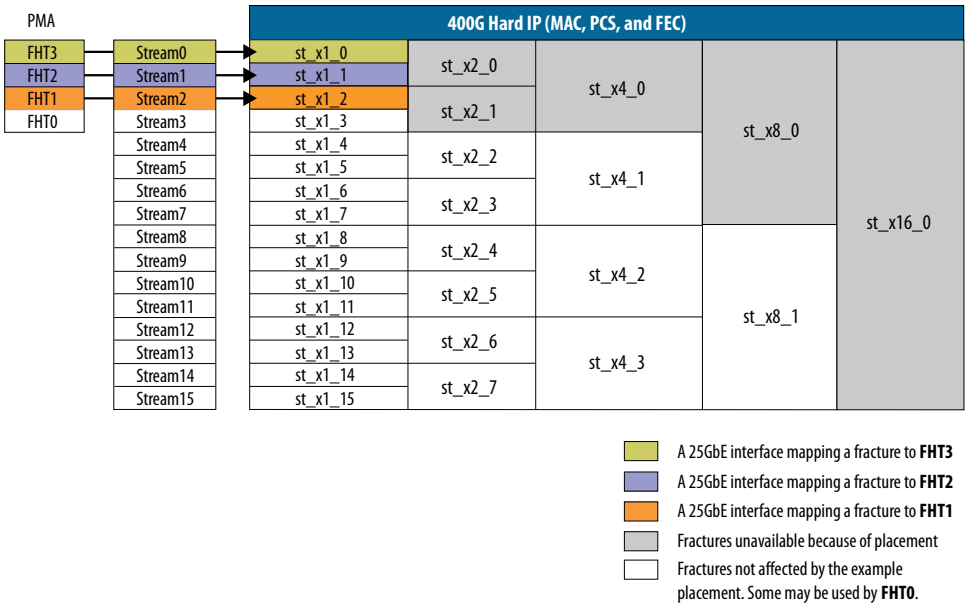
When implementing multiple interfaces, keep in mind that each fracture placement impacts the location of other fractures as shown in the following examples.

This example needs:

- Three fractures of type **st_x1** — one for each 25GbE-1 interface
- Three PMAs — one for each 25GbE-1 interface

Fracture Index	PMA Mapping
One possible fracture indices and PMA mappings:	
st_x1_0	**FHT3**
st_x1_1	**FHT2**
st_x1_2	**FHT1**

Figure 23. Implementing Three 25GbE-1 Interfaces with 400G Hard IP and FHT



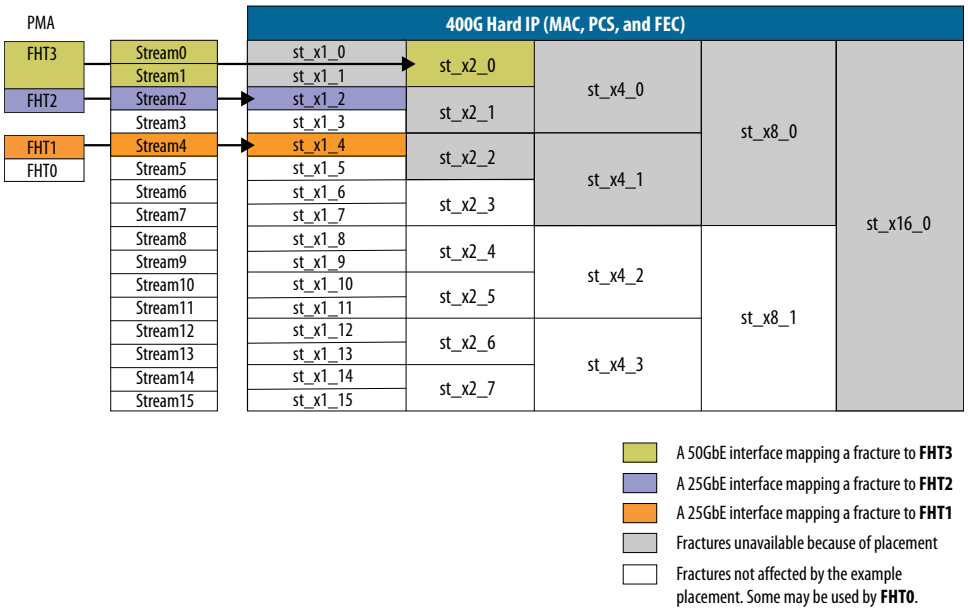
2.2.2.7. Implementing One 50GbE-1 and Two 25GbE-1 Interfaces with 400G Hard IP and FHT

This example needs:

- One fracture of type **st_x2** for the 50GbE-1 interface
- Two fractures of type **st_x1** — one for each 25GbE-1 interface
- One PMA for the 50GbE-1 interface
- Two PMAs — one for each 25GbE-1 interface

Fracture Index	PMA Mapping
One possible fracture indices and PMA mappings:	
st_x2_0	**FHT3**
st_x1_2	**FHT2**
st_x1_4	**FHT1**

Figure 24. Implementing One 50GbE-1 and Two 25GbE-1 Interfaces with 400G Hard IP and FHT



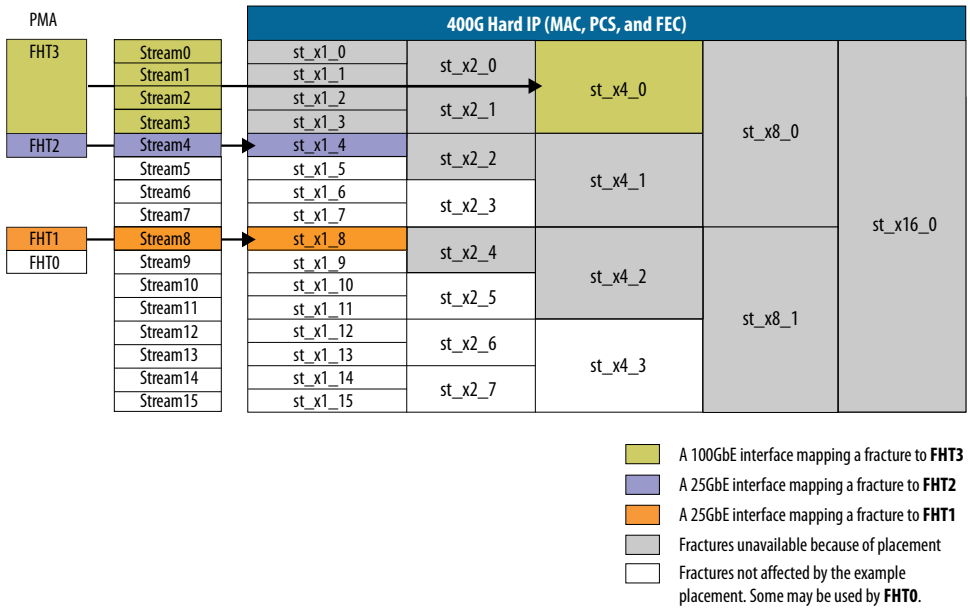
2.2.2.8. Implementing One 100GbE-1 and Two 25GbE-1 Interfaces with 400G Hard IP and FHT

This example needs:

- One fracture of type **st_x4** for the 100GbE-1 interface
- Two fractures of type **st_x1** — one for each 25GbE-1 interface
- One PMA for the 100GbE-1 interface
- Two PMAs — one for each 25GbE-1 interface

Fracture Index	PMA Mapping
One possible fracture indices and PMA mappings:	
st_x4_0	**FHT3**
st_x1_4	**FHT2**
st_x1_8	**FHT1**

Figure 25. Implementing One 100GbE-1 and Two 25GbE-1 Interfaces with 400G Hard IP and FHT



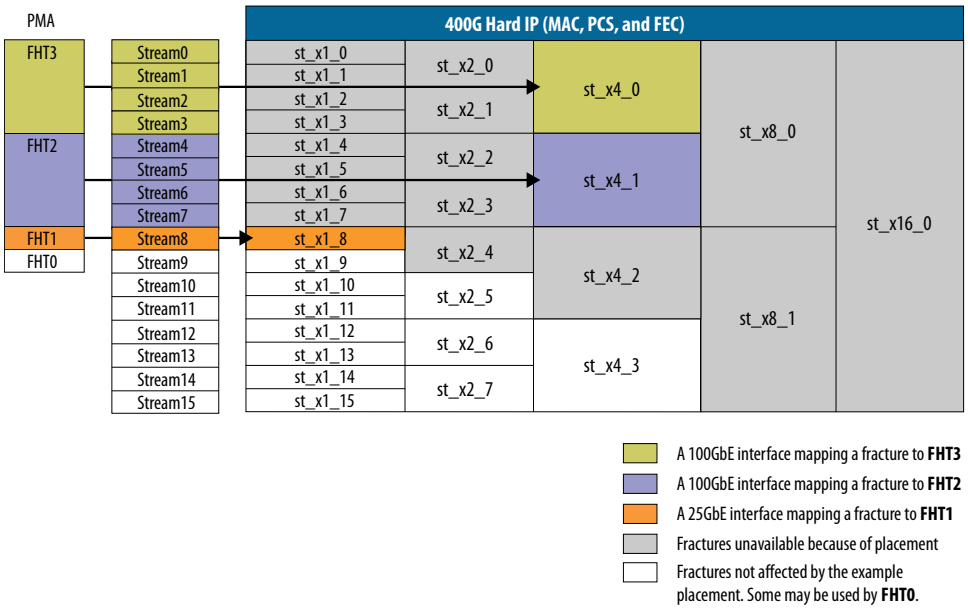
2.2.2.9. Implementing Two 100GbE-1 and One 25GbE-1 Interfaces with 400G Hard IP and FHT

This example needs:

- Two fractures of type **st_x4** — one for each 100GbE-1 interface
- One fracture of type **st_x1** for the 25GbE-1 interface
- Two PMAs — one for each 100GbE-1 interface
- One PMA for the 25GbE-1 interface

Fracture Index	PMA Mapping
One possible fracture indices and PMA mappings:	
st_x4_0	**FHT3**
st_x4_1	**FHT2**
st_x1_8	**FHT1**

Figure 26. Implementing Two 100GbE-1 and One 25GbE-1 Interfaces with 400G Hard IP and FHT



2.2.2.10. Implementing 100GbE-1, 100GbE-2, and 50GbE-1 Interfaces with 400G Hard IP and FHT

This example needs:

- One fracture of type **st_x4** for the 100GbE-2 interface
- One fracture of type **st_x4** for the 100GbE-1 interface
- One fracture of type **st_x2** for the 50GbE-1 interface
- Two PMAs for the 100GbE-2 interface
- One PMA for the 100GbE-1 interface
- One PMA for the 50GbE-1 interface

Placement Order	Fracture Index	PMA Mapping
Possible fracture index and PMA mapping option 1 (See Implementing 100GbE-1, 100GbE-2, and 50GbE-1 Interfaces with 400G Hard IP and FHT Option 1.):		
Interface 1: 100GbE-2	**st_x4_0**	**FHT3**, **FHT2**
Interface 2: 100GbE-1	**st_x4_2**	**FHT1**
Interface 3: 50GbE-1	**st_x2_6**	**FHT0**
Possible fracture index and PMA mapping option 2:		
Interface 1: 100GbE-2	**st_x4_0**	**FHT3**, **FHT2**
Interface 2: 50GbE-1	**st_x2_2**	**FHT1**
Interface 3: 100GbE-1	**st_x4_3**	**FHT0**
Unsupported fracture index and PMA mapping option 1 (See Unsupported 100GbE-1, 100GbE-2, and 50GbE-1 Interfaces Implementation with 400G Hard IP and FHT Option 1.):		
Interface 1: 100GbE-1	**st_x4_0**	**FHT3**
Interface 2: 100GbE-2	**st_x4_1**	**FHT1**, **FHT0**
Interface 3: 50GbE-1	The 50GbE-1 interface cannot be placed because there are no available **st_x2** fractures (for example, **st_x2_4** or **st_x2_5**) that can map to **FHT2**.	**FHT2**
Unsupported fracture index and PMA mapping option 2 (See Unsupported 100GbE-1, 100GbE-2, and 50GbE-1 Interfaces Implementation with 400G Hard IP and FHT Option 2 and Unsupported Crisscross Connection Implementation of 100GbE-1, 100GbE-2, and 50GbE-1 Interfaces with 400G Hard IP and FHT Option 2.):		
Interface 1: 100GbE-2	**st_x4_0**	**FHT3**, **FHT2**
Interface 2: 100GbE-1	**st_x4_2**	**FHT1**
Interface 3: 50GbE-1	**st_x2_3** This mapping is not possible because it creates a crisscross connection between the PMAs.	**FHT0**

Figure 27. Implementing 100GbE-1, 100GbE-2, and 50GbE-1 Interfaces with 400G Hard IP and FHT Option 1

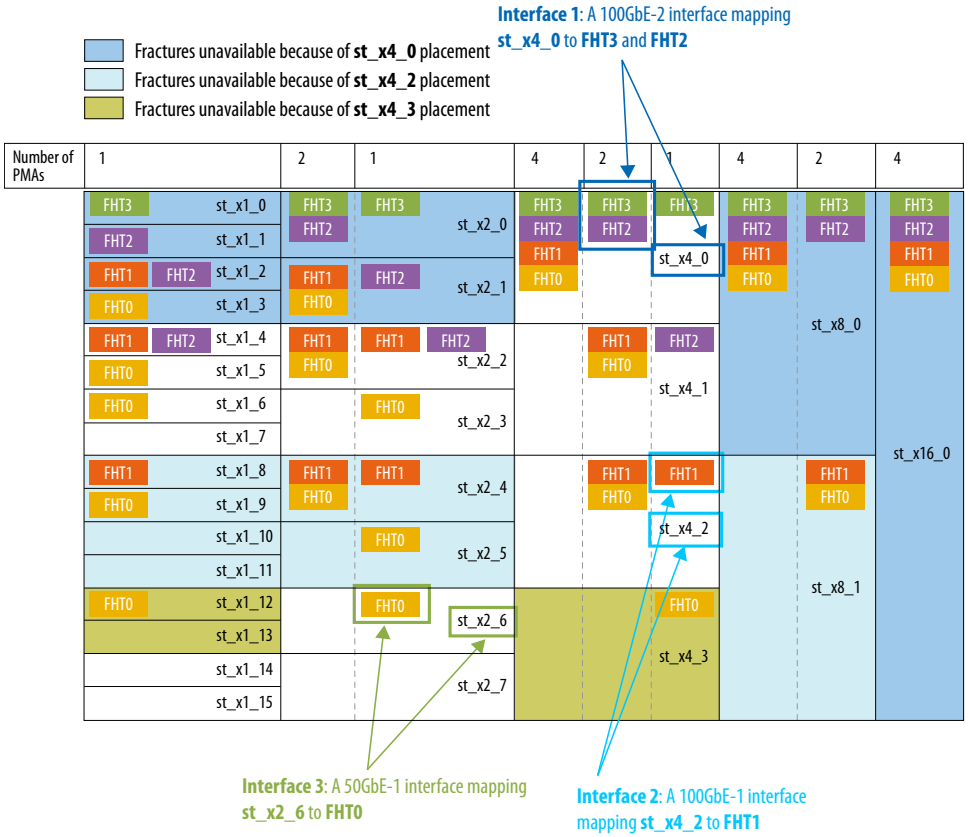


Figure 28. Unsupported 100GbE-1, 100GbE-2, and 50GbE-1 Interfaces Implementation with 400G Hard IP and FHT Option 1

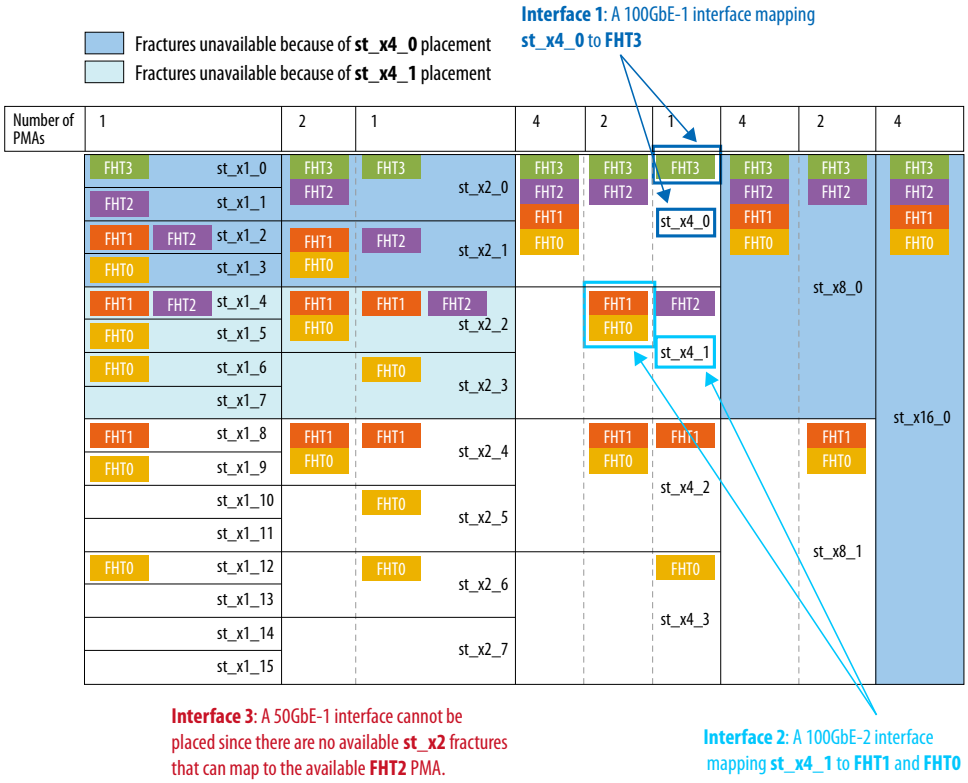


Figure 29. Unsupported 100GbE-1, 100GbE-2, and 50GbE-1 Interfaces Implementation with 400G Hard IP and FHT Option 2

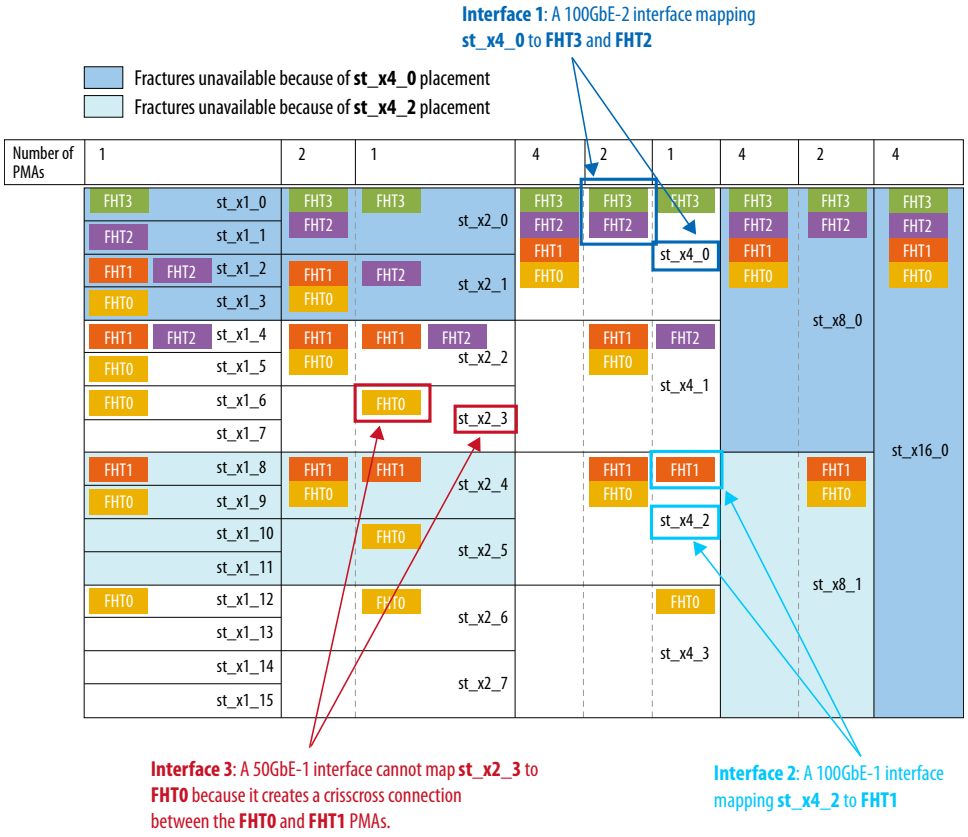
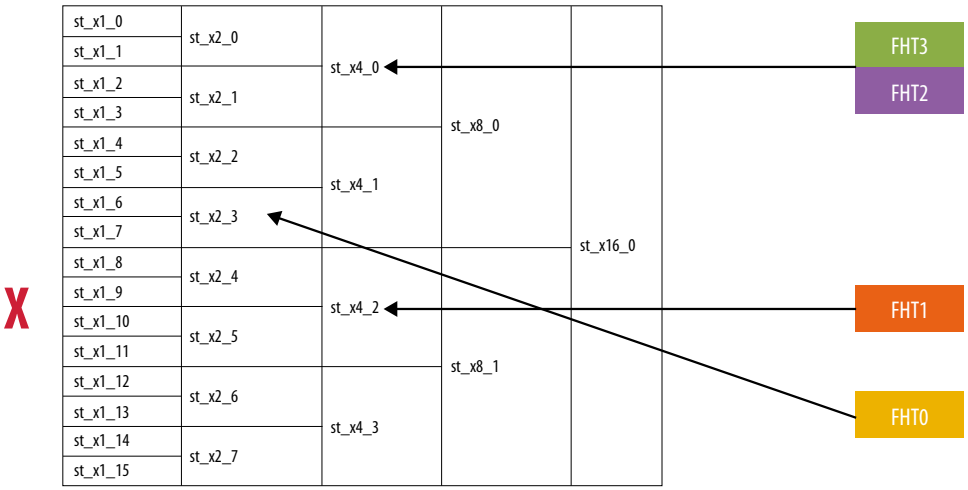


Figure 30. Unsupported Crisscross Connection Implementation of 100GbE-1, 100GbE-2, and 50GbE-1 Interfaces with 400G Hard IP and FHT Option 2



2.2.3. Hard IP Placement Rules

- 400G hard IP can access both FGT and FHT PMAs, but only one at a time.
- 200G hard IP can only access FGT PMAs (**FGT3_Quad1**-**FGT0_Quad0**).
- PCIe hard IP can only access FGT PMAs (**FGT3_Quad3**-**FGT0_Quad0**).
- FGT **Quad0** can only support 20-32 Gbps PAM4.
- If IEEE 1588 precision time protocol is enabled, **EMIB_6** and **EMIB_7** are used, so **EMIB_6** and **EMIB_7** are not available for 200G hard IP.

2.2.4. IEEE 1588 Precision Time Protocol Placement Rules

- PTP-enabled ports can be placed anywhere in 400G hard IP.
- PTP ports do not need to be adjacent to each other or to **EMIB_6** and **EMIB_7**.
- All PTP ports in an F-tile share **EMIB_6** and **EMIB_7** for timestamp information.

2.2.5. Topologies

F-tile-supported protocols use EMIBs, PMAs, and streams for the relevant hard IP. Because hard IPs in multi-protocol F-tile designs share PMAs and EMIBs, specific pairings of PMAs and EMIBs are required to support different combinations of hard IPs, PTP-enabled ports, and bandwidths. These pairings are called *topologies*. F-tile supports 15 pre-defined topologies, each with different constraints. Every F-tile design must follow one of these topologies. You cannot dynamically reconfigure from one topology to another. Dynamic reconfiguration is allowed only within a topology.

Select a topology based on the following design considerations:

- Do you need PCIe?
- Do you need IEEE 1588 precision time protocol ports?
- Do you need FHT PMA lanes?

If you need to implement multiple hard IPs, verify that there is a topology that meets your requirements.

- If your F-tile design does not utilize all tile resources, there may be more than one topology that meets your requirements.
- If more than one topology meets your requirements, select the topology with the most PMAs and streams available to ensure that the maximum number of hard IPs can be implemented.
- Use the *F-Tile Channel Placement Tool* to plan your design; it shows available PMA, stream, and EMIB locations for each topology.

Table 11. F-Tile Topologies

Topology	PCIe Hard IP		400G Hard IP					200G Hard IP		
	Avail-ability	Config-uration	Avail-ability	Configuration				Avail-ability	Configuration[4]	
				PMA	PTP	Number of PMAs	Number of Streams		Number of PMAs	Number of Streams
1	Yes	1x PCIe x16	No	N/A	N/A	N/A	N/A	No	N/A	N/A
2	Yes	2x PCIe x8	No	N/A	N/A	N/A	N/A	No	N/A	N/A
3	Yes	1x PCIe x16	Yes	FHT	Yes	4	4	No	N/A	N/A
4	Yes	4x PCIe x4	No	N/A	N/A	N/A	N/A	No	N/A	N/A
5	No	N/A	Yes	FHT	No	4	16	Yes	8	8
6	No	N/A	Yes	FHT	Yes	4	16	Yes	6	6
6a	No	N/A	Yes	FGT (4) + FHT (4)	Yes	8	16	Yes	6	6
7	Yes	1x PCIe x4	Yes	FHT	Yes	4	16	No	N/A	N/A
8	Yes	1x PCIe x8	Yes	FHT	Yes	4	10	No	N/A	N/A
9	Yes	2x PCIe x4	Yes	FHT	Yes	4	10	No	N/A	N/A
10	No	N/A	Yes	FGT	No	8	16	Yes	8	8
11	No	N/A	Yes	FGT	Yes	8	16	Yes	6	6
12	Yes	1x PCIe x8	Yes	FGT	Yes	8	11	No	N/A	N/A
13	Yes	2x PCIe x4	Yes	FGT	Yes	8	11	No	N/A	N/A
14	Yes	1x PCIe x4	Yes	FGT	Yes	8	16	No	N/A	N/A
15	No	N/A	Yes	FGT	Yes	16	16	No	N/A	N/A

For example:

- **Topology 2:** 2x PCIe x8:
  - The PCIe hard IP implements two ports of PCIe x8.
  - You cannot implement any other protocol interface in this F-tile.
  - 400G hard IP and 200G hard IP are unavailable.
- **Topology 3:** 1x PCIe x16 + 400G Hard IP (FHT) with PTP is a superset of **Topology 1:** 1x PCIe x16. That means if your target implementation works with **Topology 1:** 1x PCIe x16, it also works with **Topology 3:** 1x PCIe x16 + 400G Hard IP (FHT) with PTP.
- **Topology 6a:** Mix of Topology 6 and 11. It uses the mixed transceiver mode for 400G Hard IP with four FHT PMA and four FGT PMA (Quad2). In this topology, the 200G Hard IP supports a maximum data rate of 150G with seven FGT PMAs (Quad0: 0-3 and Quad1: 0-2).

(4) For 200G hard IP, all PMAs are FGT, and PTP is not available.

Related Information

F-Tile Channel Placement Tool

2.2.5.1. Topology 5: 400G Hard IP (FHT) + 200G Hard IP (FGT) Example

This example assumes the following design considerations for an F-tile:

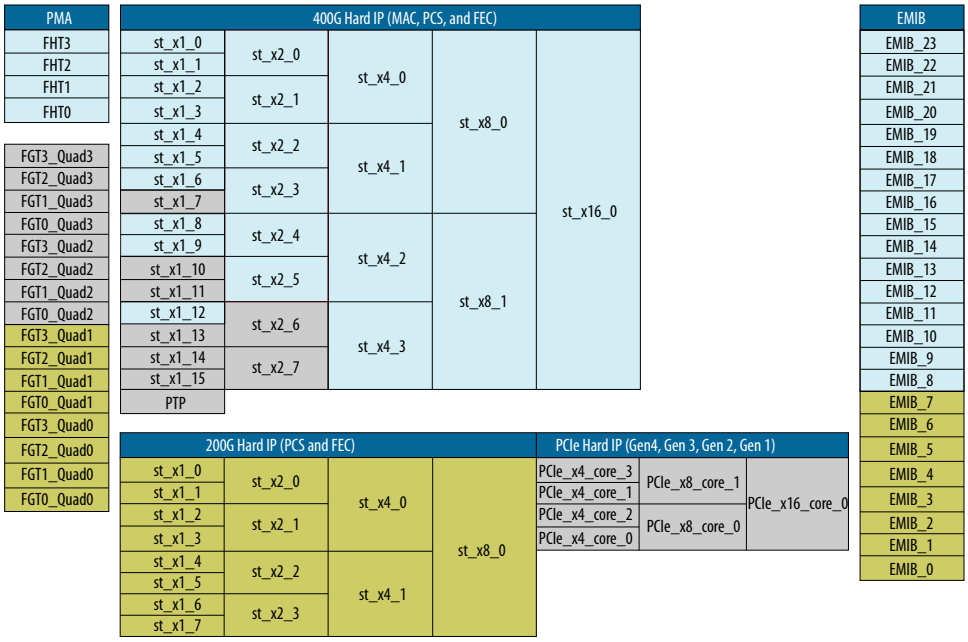
- A PCIe interface is not required.
- An IEEE 1588 precision time protocol interface is not required.
- FHT PMA lanes are required.

Topology 5: 400G Hard IP (FHT) + 200G Hard IP (FGT) is the only way to implement this design. The following figure shows the PMA, fracture, and EMIB resource availability.

Figure 31. Topology 5: 400G Hard IP (FHT) + 200G Hard IP (FGT)

Legend

- Gray: unavailable resource
- Light blue: available 400G hard IP resource
- Green: available 200G hard IP resource



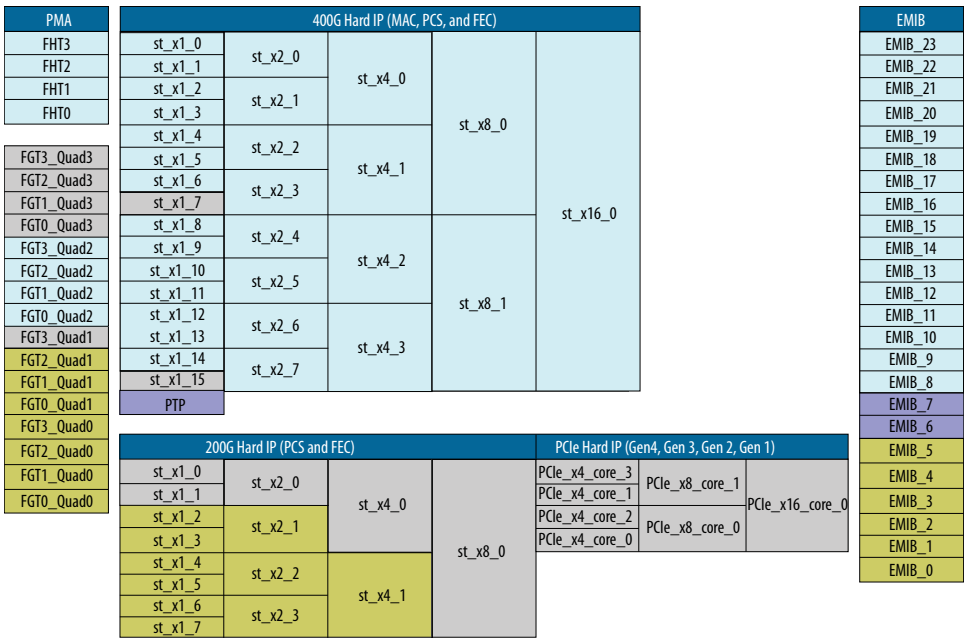
2.2.5.2. Topology 6a: 400G Hard IP with Mixed Transceiver Mode Example

Topology 6a is mix of Topology 6 and 11. It uses the mixed transceiver mode for 400G Hard IP with four FHT PMA and four FGT PMA (Quad2). In this topology, the 200G Hard IP supports a maximum data rate of 150G with seven FGT PMAs (Quad0: 0-3 and Quad1: 0-2).

Figure 32. Topology 6a: 400G Hard IP (4 FHT + 4 FGT PMAs)

Legend

- Gray: unavailable resource
- Light blue: available 400G hard IP resource
- Purple: available PTP resource
- Green: available 200G hard IP resource



2.2.5.3. Topology 14: 1x PCIe x4 + 400G Hard IP (FGT) with PTP Example

This example assumes the following design considerations for an F-tile:

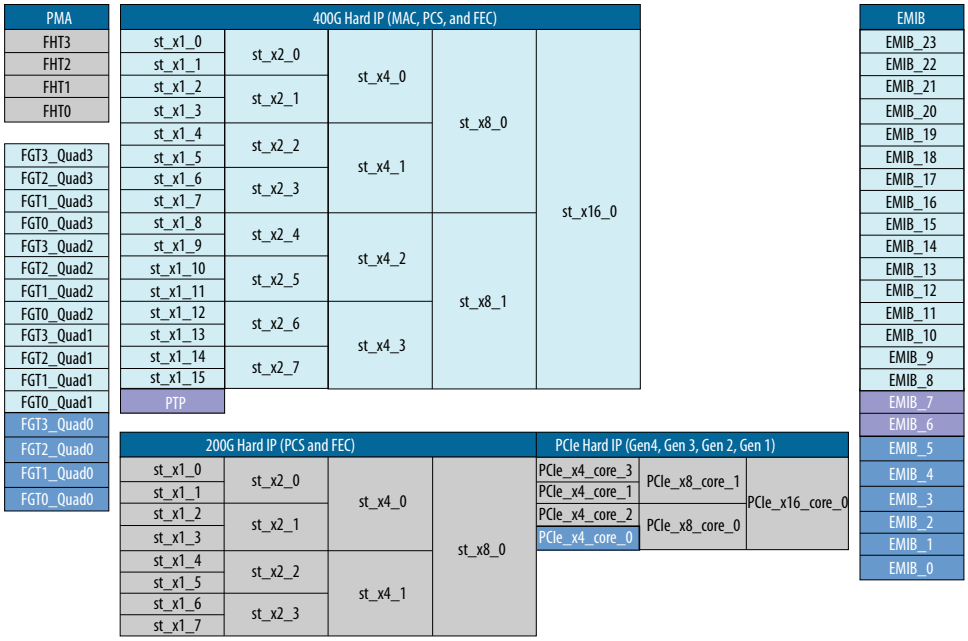
- A PCIe interface is required.
- An IEEE 1588 precision time protocol interface is required.
- FHT PMA lanes are not required.

Topology 12: 1x PCIe x8 + 400G Hard IP (FGT) with PTP, **Topology 13: 2x PCIe x4 + 400G Hard IP (FGT) with PTP**, and **Topology 14: 1x PCIe x4 + 400G Hard IP (FGT) with PTP** can implement this design. Select **Topology 14: 1x PCIe x4 + 400G Hard IP (FGT) with PTP** because it has the most available PMAs and streams. The following figure shows the PMA, fracture, and EMIB resource availability.

Figure 33. Topology 14: 1x PCIe x4 + 400G Hard IP (FGT) with PTP

Legend

- Gray: unavailable resource
- Light blue: available 400G hard IP resource
- Purple: available PTP resource
- Dark blue: available PCIe hard IP resource



2.2.6. FEC Placement Rules

- A single FEC core can be used to implement up to four different hard IP interfaces each with a different FEC type. For example, it is possible to have two RS-FEC(528, 514) mode 25GbE hard IP interfaces and one RS-FEC(544, 514) mode 50GbE hard IP interface in one FEC core. Refer to the respective hard IP protocol user guides for supported FEC types.
- All FEC-enabled hard IP interfaces in a FEC core must use the same system PLL.
- Four FEC-enabled hard IP interfaces using a **st_x1** configuration can be placed in one FEC core. The PMA location of these hard IP interfaces need not be contiguous.
- For designs using one FEC-enabled hard IP interface using a **st_x2**, **st_x4**, or **st_x8** configuration, the FEC streams must be contiguous.
- In **st_x2** configurations, the streams must be the top two or the bottom two, not the middle two, of a FEC core. Nor can they straddle FEC cores. For example, the two streams can be either **Stream0** and **Stream1** or **Stream2** and **Stream3** but not **Stream1** and **Stream2** and not **Stream3** and **Stream4**.
- All four streams in a **st_x4** configuration must be in one FEC core.
- All eight streams in a **st_x8** configuration must be in two FEC cores.

- For 200GbE and 400GbE, FEC must be enabled: LL FEC and KP FEC are the available FEC modes.
- F-tile does not allow FEC for 40GbE.
- Firecode FEC is only available for 25GbE.

Figure 34. FEC-Enabled Configuration Example with Multiple Interfaces Example 1

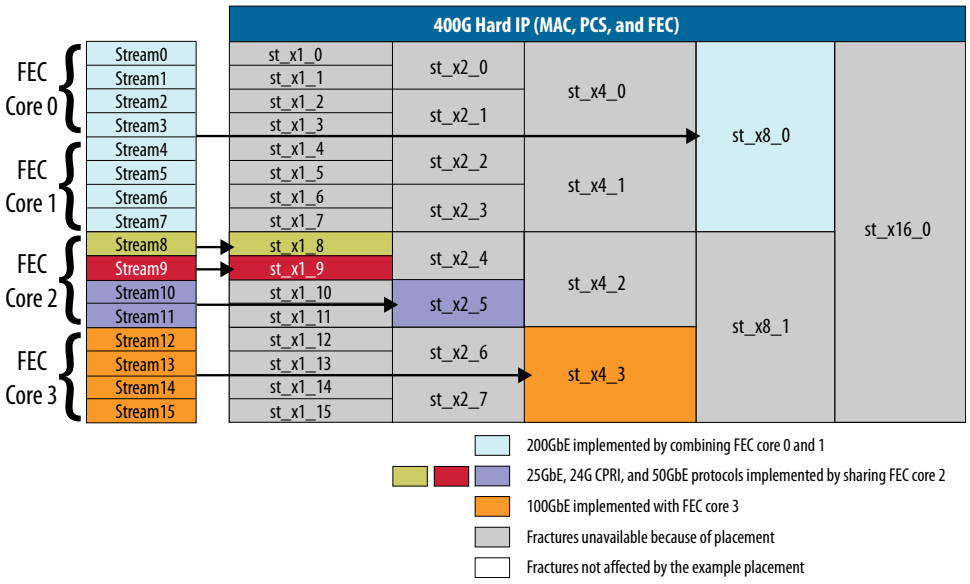
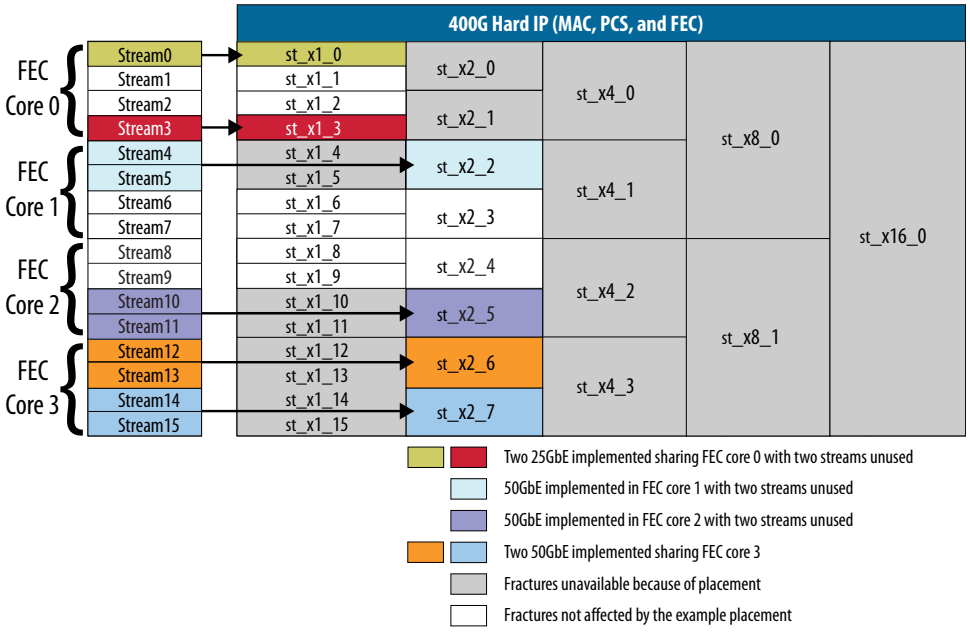


Figure 35. FEC-Enabled Configuration Example with Multiple Interfaces Example 2



2.2.7. Clock Rules and Restrictions

- When you enable the **Refclk #i is available at and after device configuration** parameter in the F-Tile Reference and System PLL Clocks Intel FPGA IP, you must have a stable and running reference clock for the system PLL and FGT PMA to configure the FPGA . Refer to Guidelines for Refclk #i is Active At and After Device Configuration for more information.

  Note: A stable reference clock implies that the reference clock meets the specifications listed in the Intel Agilex 7 FPGAs and SoCs Device Data Sheet: F-Series and I-Series.
- In oder to prevent the FHT PMA lane performance degradation, you must provide a stable and running reference clock to the FHT PMA to configure the FPGA.
- A reference clock must be up and stable before releasing the connected hard IP resets.
- 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.

  Note: After a temporary loss of the system PLL reference clock, you may observe that the first try of device reconfiguration fails. If that occurs, you should try to reconfigure the FPGA a second time.
- Once the reference clock that drives the FHT PMA is up, it must be stable; it must be present throughout the operation, must not change the frequency, and must not go down. If you are not able to adhere to this, you must reconfigure the device.
- The baud rate or line frequency of two adjacent FHT lanes must be either exactly the same (driven by the same reference clock and receiving signals from a family of transmitters synchronized to the same reference clock) or separated at least by 2,000 ppm. This is to eliminate lane-to-lane interaction.
- Each hard IP instance, for example, 25GbE with FEC, CPRI 24G with FEC, and 50GbE FEC Direct, placed in the same FEC core must use the same system PLL.
- Each Ethernet hard IP instance running IEEE 1588 precision time protocol must use the same system PLL.
- All lanes that are part of the same interface, for example, the eight lanes of a 400GbE, must use the same system PLL.
- TX simplex and RX simplex must use the same system PLL unless they are both using PMA Direct's PMA clocking mode.
- The system PLL must not be dynamically reconfigured. In addition, the system PLL input and output clock frequencies must not be dynamically reconfigured. If this is not followed, you must reconfigure the device.
- All hard IPs that are assigned to a dynamic reconfiguration group must use the same system PLL.
- System PLL clocking mode must be used for data rates between 29 Gbps and 32 Gbps NRZ.

2.2.8. Bonding Placement Rules

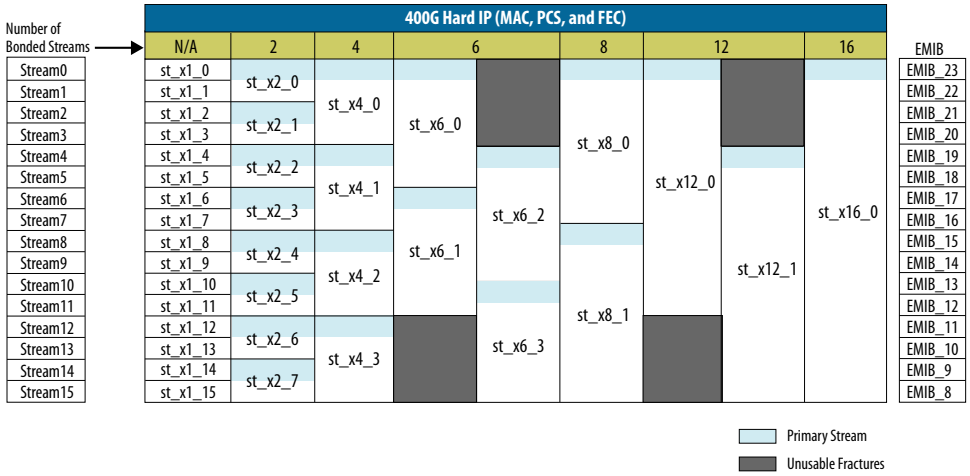
When multiple streams are bonded together, there are pre-defined, allowed locations for EMIBs and lanes based on fracture type and location.

Streams and lanes are placed contiguously in sequence. The primary stream is placed at the highest EMIB number. The secondary stream is placed at the second highest EMIB number. Subsequent streams are likewise placed in descending order.

In the following figures, **st_x6** and **st_x12** bonding placements can only be used in PMA Direct mode. These placements are primarily meant for a protocol such as JESD204B/C or protocols which require PMA Direct bonding of something other than two, four, eight, or 16 lanes.

Note: When bonding 8 PMA lanes, you must place them within two quads. When bonding 4 PMA lanes, you must place them within one quad.

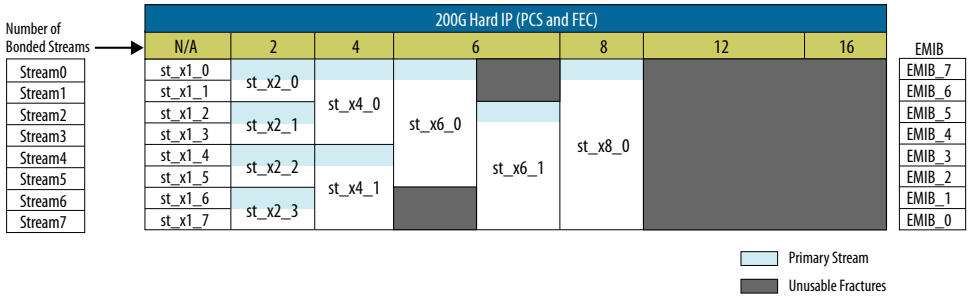
Figure 36. 400G Primary Stream Placement



If you use eight bonded TX PMA lanes in 400G hard IP with eight streams of a **st_x8** fracture:

- The eight 25 Gbps PMA lanes are denoted as `tx_serial_data[7:0]`.
- The allowed locations for primary stream `tx_serial_data[0]` are **EMIB_23** and **EMIB_15**.
- If **EMIB_15** is selected, `tx_serial_data[0]` is assigned to **FGT3_Quad1**. Subsequently, `tx_serial_data[1]` is assigned to **EMIB_14** and **FGT2_Quad1**; `tx_serial_data[2]` is assigned to **EMIB_13** and **FGT1_Quad1**, with subsequent streams following the same pattern.

Figure 37. 200G Primary Stream Placement



If you use one RX PMA lane in 200G hard IP with two streams of a **st_x2** fracture:

- The 50 Gbps PMA lane is denoted as `rx_serial_data[0]`.
- The allowed locations for primary stream `rx_serial_data[0]` are **EMIB_7** and **EMIB_5**.
- If **EMIB_7** is selected, `rx_serial_data[0]` is assigned to **FGT3_Quad1**, and the secondary stream is assigned to **EMIB_6** and **FGT2_Quad1**.

The primary lane and stream must be active before and after dynamic reconfiguration. For example, if you require an Ethernet interface that supports dynamic reconfiguration between 200G-CR4 and 100G-CR2 in 400G hard IP:

- The primary lane is denoted as `eth_tx_serial_data[0]` and `eth_rx_serial_data[0]`.
- Both **st_x8** and **st_x4** fractures are used (the former for 200G-CR4 and the latter for 100G-CR2), but not at the same time.
- **EMIB_23** and **EMIB_15** are allowed locations for the **st_x8** fracture primary stream.
- **EMIB_23**, **EMIB_19**, **EMIB_15** and **EMIB_11** are allowed locations for the **st_x4** fracture primary stream.
- Therefore, you can choose either **EMIB_23** or **EMIB_15** as the primary stream location because they are the allowed locations for both fracture types.

Use the *F-Tile Channel Placement Tool* for your channel placement planning. Use the Intel Quartus Prime **Tile Interface Planner** to sign off on the channel placement for your board layout.

Related Information

F-Tile Channel Placement Tool

2.2.8.1. Bonded Lanes Use Case 1

One 100G-4 Ethernet MAC in Ethernet hard IP with PTP enabled

- Four PMA lanes at 25.78 Gbps per PMA lane
- Modulation scheme: NRZ
- Primary streams at **EMIB_23**
- One **st_x4** fracture is used

One DisplayPort 2.0 in F-tile PMA and FEC Direct PHY IP [(5)]

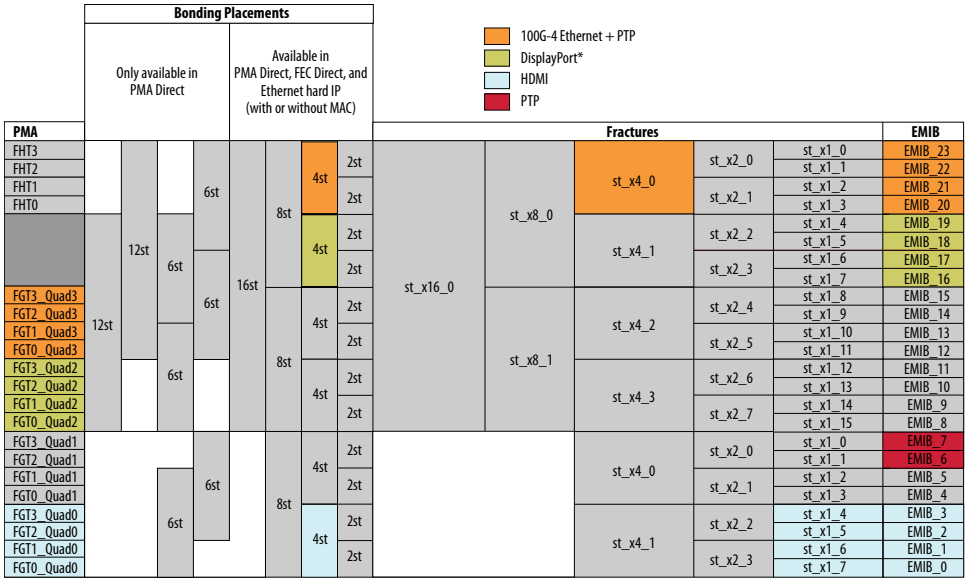
- 20.0 Gbps per PMA lane without FEC
- Dynamic lane scaling from four lanes to two lanes or one lane
- Modulation scheme: NRZ
- Primary stream at **EMIB_19**
- Four **st_x1** fractures are used

(5) IP shown for illustrative purposes. Contact Intel FPGA support for specific IP availability.

One HDMI in F-tile PMA and FEC Direct PHY IP

- 12.0 Gbps per PMA lane without FEC
- Modulation scheme: NRZ
- Primary stream at **EMIB_3**
- Four **st_x1** fractures are used

Figure 38. Bonded Lanes Use Case 1



2.2.8.2. Bonded Lanes Use Case 2

One 200G-SR4 Ethernet MAC in Ethernet hard IP

- Four PMA lanes at 53.125 Gbps per PMA lane
- Modulation scheme: PAM4
- Primary stream at **EMIB_23**
- One **st_x8** fracture is used

One 50 Gbps by three lanes in F-tile PMA and FEC Direct PHY IP

- Three PMA lanes at 50 Gbps per PMA lane with FEC
- Modulation scheme: PAM4
- Primary stream at **EMIB_15**
- Three **st_x2** fractures are used

One 50G-CR1 Ethernet PCS in Ethernet hard IP

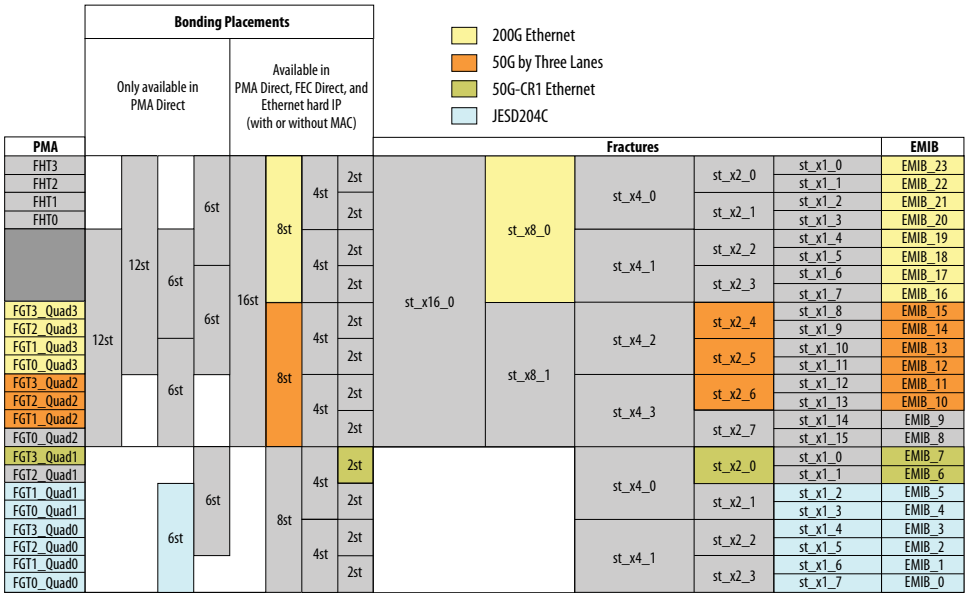
- One PMA lane at 53.125 Gbps
- Modulation scheme: PAM4
- Primary stream at **EMIB_7**
- One **st_x2** fracture is used

One JESD204C by six lanes in F-tile PMA and FEC Direct PHY IP [6]

- Six PMA lanes at 32.0 Gbps per PMA lane without FEC
- Modulation scheme: NRZ
- Primary stream at **EMIB_5**
- Six **st_x1** fractures are used

Figure 39. Bonded Lanes Use Case 2

The **st_x2_7** fracture is reserved for 50 Gbps PMA and FEC Direct PHY IP. This fracture is associated with **EMIB_8** and **EMIB_9**. This fracture and its associated EMIBs are unavailable to any other high-speed serial link IP. **FGT2_Quad1**, **FGT0_Quad2**, and all FHT PMA lanes are not available because there are no EMIBs available.



2.2.9. Preserving Unused PMA Lanes

You must preserve the unused FHT and FGT PMA lanes that you plan to use later in your design to ensure no degradation in the PMA lane performance or maximum data rate during the period of use.

To preserve the performance of unused PMA lanes, the Intel Quartus Prime software can program the unused PMA lanes, such that the analog circuitry in their transmit and receive stages toggles at a low data rate.

Unused PMA lanes may appear in any of the following ways in the F-tile:

- Unused PMA lanes in a completely unused F-tile.
- Unused PMA lanes in a partially used F-tile.

[6] IP shown for illustrative purposes. Contact Intel FPGA support for specific IP availability.

Unused PMA Lanes in a Completely Unused F-tile

You must either preserve the completely unused F-tile with `.qsf` assignments or ground the power rails. If you do not plan to use the F-tile in the future and do not want to preserve the PMA lanes:

- You must tie the various F-tile power rails to ground to save power.
- You must not use `.qsf` assignments shown below in your project, to preserve the F-tile.

If you do not follow these recommendations, it can result in a configuration error.

To preserve a completely unused F-tile to use it later:

- You must configure and power the F-tile and connect all power rails to the appropriate power supplies.
- You must use `.qsf` assignments in your project to preserve the unused F-tile.

If you do not follow these recommendations, it can result in a configuration error.

You must use one of the `.qsf` assignments shown below to preserve unused lanes in the F-tile.

To preserve all unused PMA lanes in a single F-tile in a package, use the following single pin F-tile `.qsf`:

```
set_instance_assignment -name PRESERVE_UNUSED_XCVR_CHANNEL ON -to <pinname>
```

Example:

```
set_instance_assignment -name PRESERVE_UNUSED_XCVR_CHANNEL ON -to JW83
```

- *<pinname>* identifies the corresponding unused F-tile for preservation.
- Preserves the entire F-tile with a single pin. Pin can be specified on any lane, even if you do not connect the pin on the board.
- You can also use this `.qsf` assignment multiple times with corresponding pins from each F-tile to preserve multiple unused F-tiles.

If you have multiple unused tiles (including all unused F-tiles and other tiles such as R-tiles in a package), you can use the following global `.qsf` assignment to preserve all unused PMA lanes in all unused tiles in a package:

```
set_global_assignment -name PRESERVE_UNUSED_XCVR_CHANNEL ON
```

Note: Do not use this `.qsf` assignment if you do not need to preserve all the unused tiles and have tied the power rails to ground for some of the tiles. This can result in a configuration error.

Example cases:

- You have 4 tiles in a package – one partially used tile and three others you want to preserve. You can use the global `.qsf` assignment to preserve the three tiles.
- You have 4 tiles in a package – one partially used tile, one tile with the power rails tied to ground to save power and two other tiles that you want preserve. Do not use the global `.qsf` assignment, but instead you must use the single pin F-tile `.qsf` assignment to preserve those two tiles.

Unused PMA lanes in a Partially Used F-tile

If your design does not instantiate (does not use) a PMA lane, preservation of the unused PMA lane in the partially used F-tile takes place by default.

If you instantiate a PMA lane in your partially used F-tile design for future use, you must fulfill the following conditions:

1. If the PMA reference clock is not available, then the PMA must be held in reset before disconnecting the reference clock. For example, when you are using the HDMI IP.
2. You must not send long periods of all zeros or all ones on the TX PMA lane. If the PMA is held in reset, you do not need to follow this rule.
3. For the FHT PMA lanes, you must set `cfg_preserve_enable` (0xF0030[3:0]) to `4'b1111` to preserve the lanes. LSB is for lane 0 and MSB is for lane 3. Refer to *FHT PMA Register Map* to access the `cfg_preserve_enable` register.

2.3. PMA Architecture

Two PMA types are supported: FHT and FGT. See *PMA Data Rates* for the data rates supported by each.

Related Information

PMA Data Rates on page 14

2.3.1. FHT PMA Architecture

The FHT PMA supports the following parallel data widths.

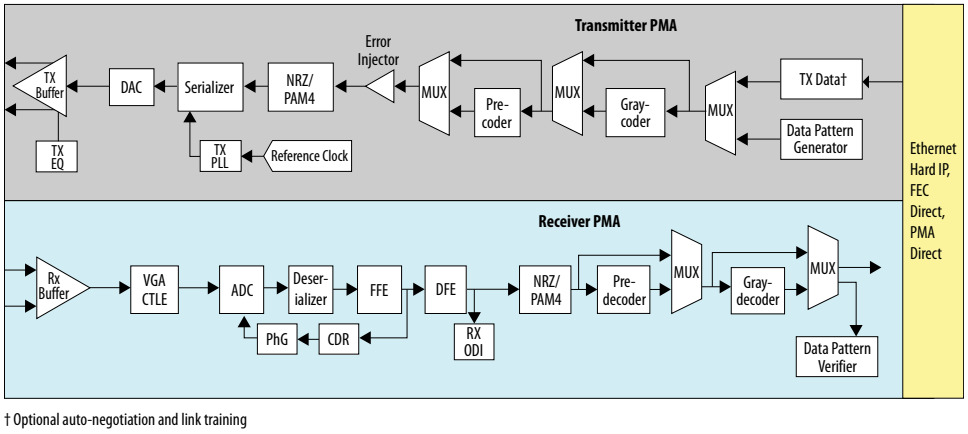
Table 12. FHT PMA Data Widths

PMA Width	Modulation	Supported Data Rates
32	NRZ	24 - 29 Gbps
64	NRZ and PAM4	48 - 58 Gbps
128	PAM4	96 - 116 Gbps

Figure 40. FHT PMA Block Diagram

Legend:

- analog-to-digital converter (ADC)
- clock data recovery (CDR)
- continuous time-linear equalization (CTLE)
- decision feedback equalization (DFE)
- digital-to-analog converter (DAC)
- feed forward equalization (FFE)
- phase generator (PhG)
- serial-in, parallel-out (SIPO)
- transmitter buffer (TX Buffer)
- transmitter equalizer (TX EQ)
- voltage gain amplifier (VGA)



† Optional auto-negotiation and link training

2.3.1.1. FHT Transmitter PMA Architecture

2.3.1.1.1. FHT Transmitter Buffer and Phase Generator

The transmitter differential I/O buffer converts the serialized bit stream to an electrical signal suitable for transmission across a cable or PCB trace. The main cursor, pre-cursor taps, and post-cursor taps of the FFE circuit control the transmitter swing strength and shape the transmitter output waveform.

The transmitter buffer can be programmed to support the taps listed in the following table. Choose the frequency response of the filter to compensate for lane impairments such as intersymbol interference (ISI), crosstalk, frequency-dependent losses, and reflections.

Table 13. FHT Transmitter PMA Equalizer Parameters for NRZ and PAM4 Modes

Parameter (Cursor)	Description	Rule			Increment and Decrement Size
		Minimum	Default	Maximum	
C_{-3}	Pre-cursor 3	-8	0	+7.75	0.25
C_{-2}	Pre-cursor 2	-8	0	+7.75	0.25
C_{-1}	Pre-cursor 1	-16	0	+15.5	0.5
C_0	Main cursor	0	41.5	+41.5(7)	0.5
C_{+1}	Post-cursor 1	-16	0	+15.5	0.5
C_{+2}	Post-cursor 2	-8	0	+7.75	0.25
C_{+3}	Post-cursor 3	-8	0	+7.75	0.25
C_{+4}	Post-cursor 4	-8	0	+7.75	0.25

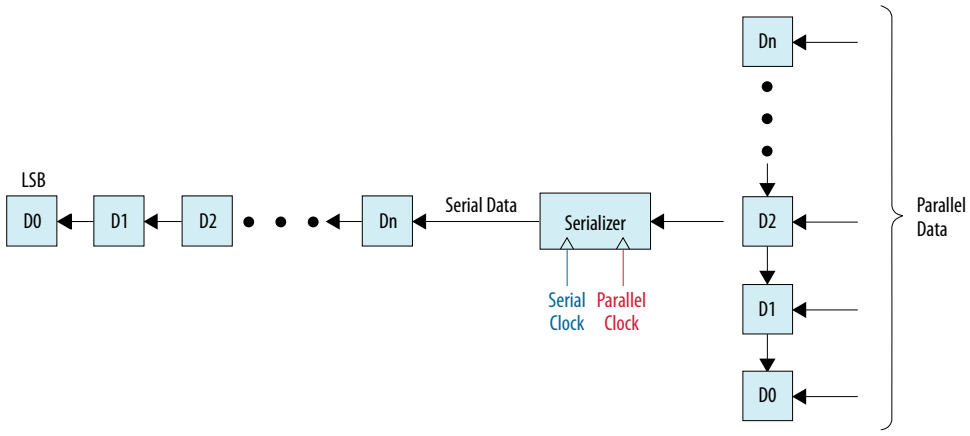
For the most capable transmitter equalizer and best amplitude control, where ABS is the absolute value:

- $ABS(C_{-3}) + ABS(C_{-2}) + ABS(C_{-1}) + C_0 + ABS(C_{+1}) + ABS(C_{+2}) + ABS(C_{+3}) + ABS(C_{+4}) \leq 41.5$
- $ABS(C_{-3}) + ABS(C_{-2}) + ABS(C_{-1}) + ABS(C_{+1}) + ABS(C_{+2}) + ABS(C_{+3}) + ABS(C_{+4}) \leq C_0$

2.3.1.1.2. FHT Serializer

The serializer converts the received parallel data into a serial data stream. The serializer supports the following serialization factors: 32, 64, and 128. The serializer is hard-coded to the least significant bit first.

Figure 41. Serializer



2.3.1.1.3. FHT Gray-Coder and Pre-Coder

PAM4 patterns generated in PAM4 mode are gray-encoded by default. For NRZ signals, the gray-coder is bypassed.

(7) Provides the maximum TX output swing

Table 14. Gray-Coder

Linear	Gray
00	00
01	01
10	11
11	10

The transmitter also includes an optional pre-coder that you can enable for both PAM4 and NRZ signals. Once you turn it on, it performs 1÷(1+D) encoding on all data bits until you disable it.

For the FHT gray-decoder and pre-decoder, reverse the process for the FHT gray-coder and pre-coder.

2.3.1.1.4. FHT Data Pattern Generator and Verifier

The data pattern generator is a design-for-test feature for generating PHY data traffic. This feature allows you to debug the PMA without involving the upper protocol hard IP layers. F-tile has a pseudo-random binary sequence (PRBS) pattern generator on the PMA which operates in all bit modes and can generate several patterns. The pattern and size are programmable.

There are patterns supporting both NRZ and PAM4. PRBS NRZ patterns are different from PAM4 patterns. For the same setting, depending on the encoding mode, either PRBSx (NRZ) or PRBSxQ (PAM4) is configured. Different specifications such as CEI OIF and IEEE 803.2 refer to quaternary PAM4 patterns differently. For example, QPRBS13 is identical to PRBSQ13 and QPRBS31 is identical to PRBSQ31.

Table 15. Supported Programmable PRBS Patterns by Mode

NRZ Mode	PAM4 Mode
PRBS7 PRBS9 PRBS11 PRBS13 PRBS23 PRBS31[8] PRBS58	PRBS7Q PRBS9Q PRBS11Q PRBS13Q PRBS23Q PRBS31Q PRBS58Q
User-defined pattern (32 bit, 64 bit, or 128 bit)	
Alternating 0s and 1s pattern (repeat one, eight, or 64 times)	

The data pattern verifier functions like the generator.

(8) The **PRBS31**, **QPRBS13**, **PRBS13Q**, **PRBS31Q**, **SSPR**, **SSPR1**, and **SSPRQ PRBS generator mode** settings are not currently supported through the IP GUI, although present in the parameter editor. Do not select any of the unsupported **PRBS generator mode** settings. Specify these settings using registers.

2.3.1.2. FHT Receiver PMA Architecture

The receiver recovers the clock information from the received serial data, deserializes the high-speed serial data, and creates a parallel data stream for either the receiver Ethernet hard IP, FEC block, or FPGA core.

2.3.1.2.1. FHT Receiver Buffer and Equalizer

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) on every stream for optimum performance, receiver equalization is self-triggered, requires no input, and is independent of system initial conditions.

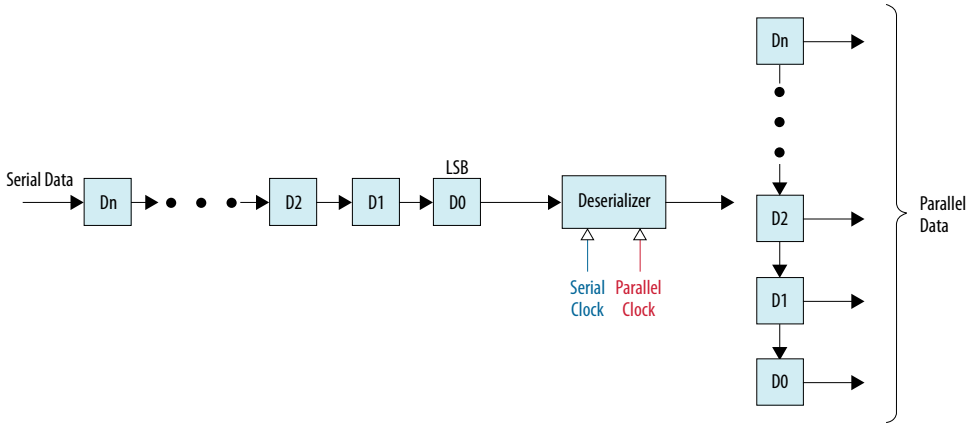
2.3.1.2.2. CDR Block

Clocking resources in the receiver enable the clock data recovery feature. The CDR block locks to the received signal and extracts the transmitted data sequence by recovering the clocking information from the distorted received signal.

2.3.1.2.3. FHT 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 or FPGA core. The deserializer supports the following deserialization factors: 32, 64, and 128.

Figure 42. Deserializer



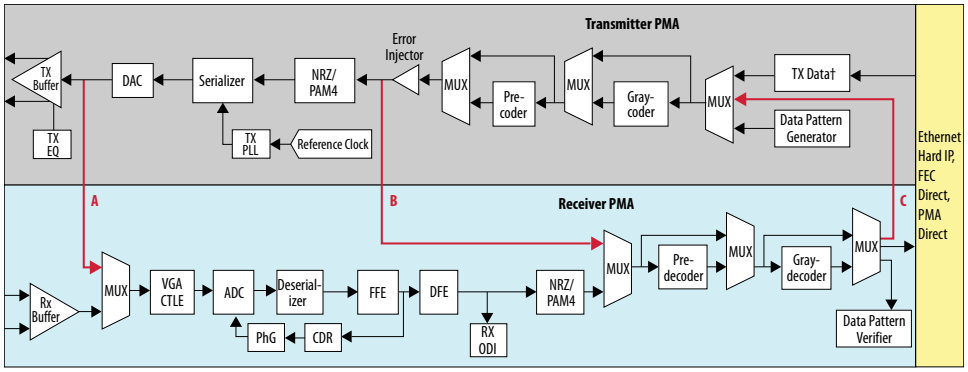
2.3.1.3. FHT PMA Loopback Modes

Loopback modes are design-for-test features that verify different blocks of the PMA. FHT PMAs have loopback modes to debug different blocks of the PMA.

Note: PMA-receiver-to-transmitter reverse serial loopback is not supported.

Figure 43. FHT Loopback Modes

- A PMA-transmitter-to-receiver internal serial loopback
- B PMA-transmitter-to-receiver digital parallel loopback
- C PMA-receiver-to-transmitter reverse parallel loopback



† Optional auto-negotiation and link training

PMA-Transmitter-to-Receiver Internal Serial Loopback

The internal serial loopback path sets the CDR to recover data from the serializer instead of the receiver serial input pin. The transmitter buffer sends data normally, but internal serial loopback takes the data before the buffer. It is implemented completely on the PMA and does not require any connector on the serial path.

The FHT lane also supports external connectivity to connect the transmitter differential outputs to the receiver differential inputs.

- Mission mode is an external connection where the data source is something other than the FHT transmitter.
- External connectivity supports, for example, a bit-error rate tester (BERT) or another device's transmitter providing data to the FHT receiver.

PMA-Transmitter-to-Receiver Digital Parallel Loopback

In a digital parallel loopback path, the parallel data stream of the transmitter is looped back as the parallel data input stream for the receiver.

PMA-Receiver-to-Transmitter Reverse Parallel Loopback

The reverse parallel loopback path sets the transmitter buffer to transmit data fed directly from the CDR recovered data. From an external instrument, data is fed to the receiver buffer, and the deserialized parallel data stream of the receiver is looped back as the parallel data input stream for the transmitter.

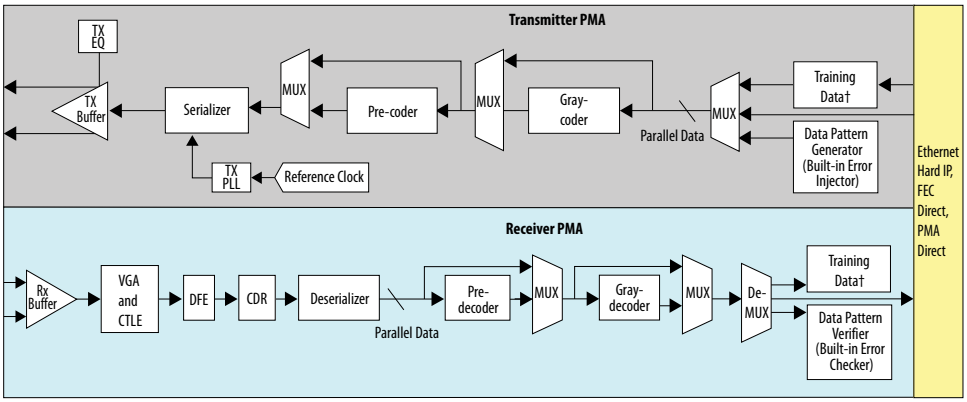
2.3.2. FGT PMA Architecture

The FGT PMA supports the following parallel data widths:

Table 16. FGT PMA Data Widths

		Supported Data Rates	
PMA Width	**Modulation**	**PMA Clocking (906.25 MHz Max)**	**System PLL Clocking (1 GHz Max)**
8	NRZ	1-7.25 Gbps	1-8 Gbps
10	NRZ	1-9.0625 Gbps	1-10 Gbps
16	NRZ	1-14.5 Gbps	1- 16 Gbps
20	NRZ	1-18.1250 Gbps	1-20 Gbps
32	NRZ and PAM4	1-29 Gbps	1-32 Gbps
64	PAM4	>32-58 Gbps	>32-58.125 Gbps

Figure 44. FGT PMA Block Diagram



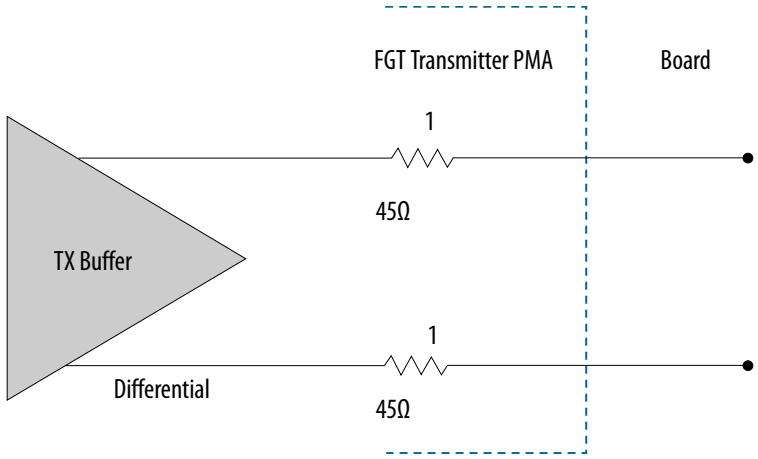
† Optional auto-negotiation and link training

2.3.2.1. FGT Transmitter PMA Architecture

2.3.2.1.1. FGT Transmitter Buffer and Phase Generator

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

Figure 45. 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 17. FGT Transmitter PMA Equalizer Parameters for NRZ and PAM4 Modes

Note: Applicable for Intel Agilex 7 F-tile devices with OPNs mentioned in Appendix A.1.

Register Value	QSF Parameter	Cursor	Rule			Increment and Decrement Size
			Minimum	Default	Maximum	
pre_tap_2	txeq_pre_tap_2	C_{-2}	0	0	+7	1.0
pre_tap_1	txeq_pre_tap_1	C_{-1}	0	0	+15	1.0
main_tap	txeq_main_tap	C_0 (9)	0	0	+55 (10)	1.0
post_tap_1	txeq_post_tap_1	C_{+1}	0	0	+19	1.0

(9) C0 = main_tap + 1 - pre_tap_1 -pre_tap_2 - post_tap_1 for Intel Agilex 7 F-tile devices with OPNs mentioned in *Appendix A.1*.

(10) main_tap and txeq_main_tap maximum value for Intel Agilex 7 F-tile devices with OPNs mentioned in *Appendix A.1*.

Table 18. FGT Transmitter PMA Equalizer Parameters for NRZ and PAM4 Modes

Note: Applicable for Intel Agilex 7 F-tile ES devices, excluding OPNs mentioned in *Appendix A.1*.

Register Value	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[11]	0	+47[12]	1.0
post_tap_1	C_{+1}	0	+19	1.0

The transmitter buffer equalizer parameter combinations follow the rules shown below.

- For Intel Agilex 7 F-tile ES devices, excluding OPNs mentioned in *Appendix A.1*:
  1. main_tap - 2×pre_tap_1 - 2×post_tap_1 ≥ 5
  2. (main_tap + 9 - 2×pre_tap_1 - pre_tap_2 - 2×post_tap_1) ÷ (main_tap + 9 - pre_tap_2 - 2×post_tap_1) > 0
  3. (main_tap + 9 - 2×pre_tap_1 - pre_tap_2 - 2×post_tap_1) ÷ (main_tap + 9 - 2×pre_tap_1 - pre_tap_2) > 0
  4. QSF: (txeq_main_tap + txeq_pre_tap_1 + txeq_pre_tap_2 + txeq_post_tap_1) ≤ 47
- For Intel Agilex 7 F-tile devices with OPNs mentioned in *Appendix A.1*:
  1. main_tap - 2×pre_tap_1 - 2×post_tap_2 ≥ 13
  2. (main_tap + 1 - 2×pre_tap_1 - pre_tap_2 - 2×post_tap_1) ÷ (main_tap + 1 - pre_tap_2 - 2×post_tap_1) > 0
  3. (main_tap + 1 - 2×pre_tap_1 - pre_tap_2 - 2×post_tap_1) ÷ (main_tap + 1 - 2×pre_tap_1 - pre_tap_2) > 0

Related Information

F-Tile TX Equalizer Tool on page 221

2.3.2.1.2. FGT Serializer

The serializer converts the received parallel data into a serial data stream. The serializer supports the following serialization factors: 32, 64. The serializer is hard-coded to the least significant bit first.

Related Information

FHT Serializer on page 57

(11) C0 = main_tap + 9 - pre_tap_1 - pre_tap_2 - post_tap_1 for Intel Agilex 7 F-tile ES devices, excluding OPNs mentioned in *Appendix A.1*.

(12) main_tap maximum value for Intel Agilex 7 F-tile ES devices, excluding OPNs mentioned in *Appendix A.1*.

2.3.2.1.3. FGT Gray-Coder and Pre-Coder

Output lanes transmitting PAM4 map to gray-coded, pre-coded, or gray-coded and pre-coded PAM4 symbols. For NRZ signals, the gray-coder and pre-coder are bypassed.

For the FGT gray-decoder and pre-decoder, reverse the process for the FGT gray-coder and pre-coder.

2.3.2.1.4. FGT Data Pattern Generator and Verifier

The PMA supports a built-in transmitter data pattern generator for transmit characterization. The pattern and size are programmable.

Table 19. Supported Programmable PRBS Patterns by Mode

NRZ	PAM4
PRBS7 PRBS9 PRBS10 PRBS13 PRBS15 PRBS23 PRBS28 PRBS31[13]	QPRBS13 QPRBS31 PRBS13Q PRBS31Q SSPR SSPR1 SSPRQ
User-defined pattern (up to 320 bits)	

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:

- PRBS13Q, PRBS31Q are defined in IEEE Std 802.3bs-2017.
- QPRBS13 is defined in IEEE Std 802.3 94.2.9.3.
- SSPR is Short Stress Pattern Random test pattern. It is defined in OIF CEI 3.1 2.D.2
- SSPR1 is defined in OIF CEI 3.1 2.D.2.
- SSPRQ is defined in IEEE 802.3 120.5.11.2.3.

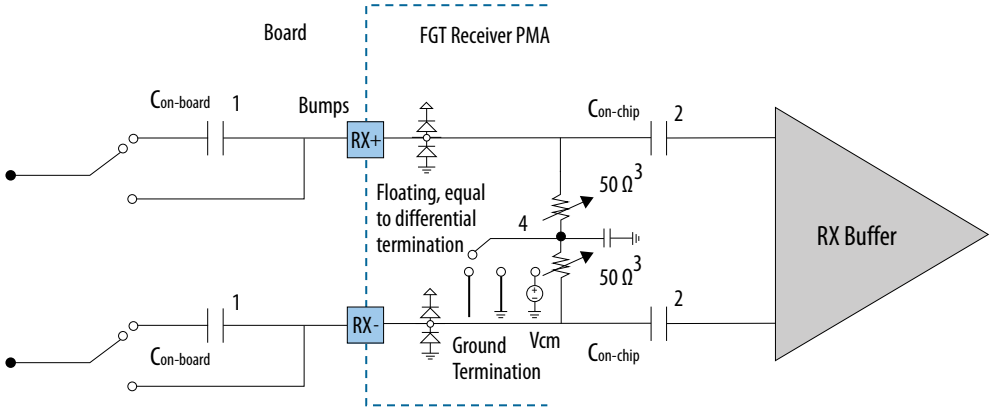
2.3.2.2. FGT Receiver PMA Architecture

2.3.2.2.1. FGT Receiver Buffer and Equalizer

A simplified FGT receiver analog front end is shown in the following figure.

(13) The **PRBS31**, **QPRBS13**, **PRBS13Q**, **PRBS31Q**, **SSPR**, **SSPR1**, and **SSPRQ PRBS generator mode** settings are not currently supported through the IP GUI, although present in the parameter editor. Do not select any of the unsupported **PRBS generator mode** settings. Specify these settings using registers.

Figure 46. Simplified RX 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. For example, PCIe requires 176nF to 265nF on board AC coupling capacitors.
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 50Ω single ended. It 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 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.

The receiver buffer and equalizer function the same for FHT and FGT PMAs. See *FHT Receiver Buffer and Equalizer* for details.

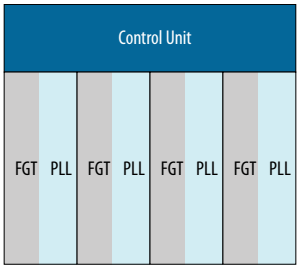
Related Information

FHT Receiver Buffer and Equalizer on page 59

2.3.2.2.2. 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.

Figure 47. Control Unit



2.3.2.2.3. FGT 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 or FPGA core. The deserializer supports the following deserialization factors: 32, 64.

Related Information

FHT Deserializer on page 59

2.3.2.3. FGT PMA Tuning

The FGT PMA supports automatic RX adaptation to meet the most advanced PAM4 and NRZ protocols, across temperature variations. The FGT PMA also supports manual settings for tuning. The FGT PMA uses automatic RX adaptation unless you add an assignment to the project `.qsf` file

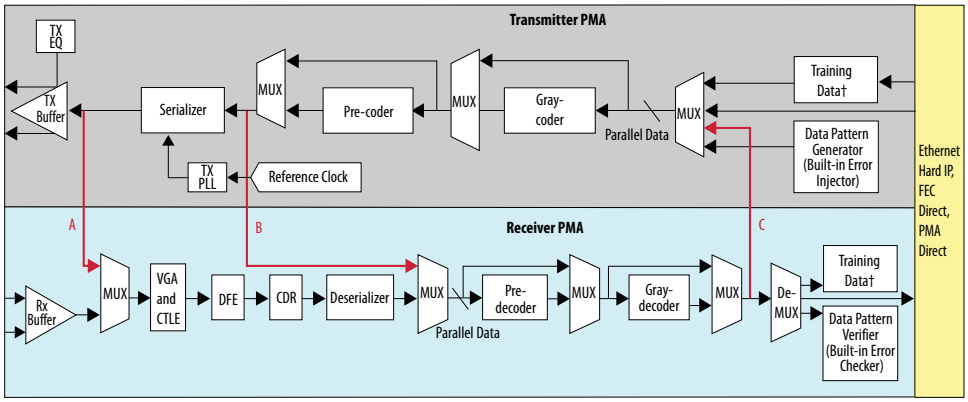
2.3.2.4. FGT PMA Loopback Modes

The PHY contains multiple parallel, serial data, and clock loopbacks across PHY interfaces for BIST. These loopbacks provide support for multiple PHY configurations.

Figure 48. FGT PMA Loopback Modes

The IP parameter editor does not currently support the loopback modes. Use register settings to specify loopback modes.

- **A. PMA-Transmitter-to-Receiver Internal Serial Loopback**: Loops back the transmitter pre-driver differential I/O signals to the midpoint of the receiver equalizer. The internal serial loopback path sets the CDR to recover data from the serializer instead of the receiver serial input pin. The transmitter buffer sends data normally, but the internal serial loopback takes the data before the buffer. It is implemented completely in the PMA and does not require any connector on the serial path.
- **B. PMA-Transmitter-to-Receiver Digital Parallel Loopback**: Parallel loopback from the PMA transmit lane 64 bit data ports to the receive lane 64 bit data ports. In a digital parallel loopback path, the parallel data stream of the transmitter is looped back as the parallel data input stream for the receiver.
- **C. PMA-Receiver-to-Transmitter Reverse Parallel Loopback**: Parallel loopback from the PMA receive lane 64 bit data ports to the transmit lane 64 bit data ports. The reverse parallel loopback path sets the transmitter buffer to transmit data fed directly from the CDR recovered data. From an external instrument, data is fed to the receiver buffer, and the deserialized parallel data stream of the receiver is looped back as the parallel data input stream for the transmitter.



† Optional auto-negotiation and link training

2.4. Clock Architecture

2.4.1. Reference Clock Network

There are two reference clocks for FHT and ten reference clocks for FGT (eight shared with system PLLs). FHT and FGT reference clocks are not shared.

2.4.1.1. FHT Reference Clock Network

There are two reference clocks (`refclk[0]` and `refclk[1]`) for FHT PMAs, both accessible by any of the four FHT PMAs. `refclk[0]` and `refclk[1]` can be different frequencies. The frequency range is 100-200 MHz, and it is a continuous range.

As shown in FHT Reference Clock Network, FHT has six PLLs.

- Two common PLLs connected to all four lanes: PLL A and PLL B
- Four lane PLLs, one PLL per lane for all four lanes: TX PLL

To optimize performance, FHT has a cascaded PLL scheme between common PLLs and lane PLLs, the common PLLs providing a cleaner clock to the lane PLLs.

- Common PLLs and lane PLLs support integer and fractional modes. However, a common PLL cannot be in fractional mode when the corresponding lane PLL is in fractional mode. See the following table.
- One common PLL drives the microcontroller. The reference clock that drives this common PLL must be present and stable throughout F-tile operation.
- Common PLLs generate two clock frequencies (100 and 156.25 MHz) which go to the lane PLLs. One of them must be selected to drive the lane PLLs.

Table 20. Supported Combinations of Lane PLL and Common PLL Modes

Lane PLL Mode	Corresponding Common PLL Mode	Supported or Not Supported
Integer	Integer	Supported
Integer	Fractional	Supported
Fractional	Integer	Supported
Fractional	Fractional	Not supported

Figure 49. FHT Reference Clock Network

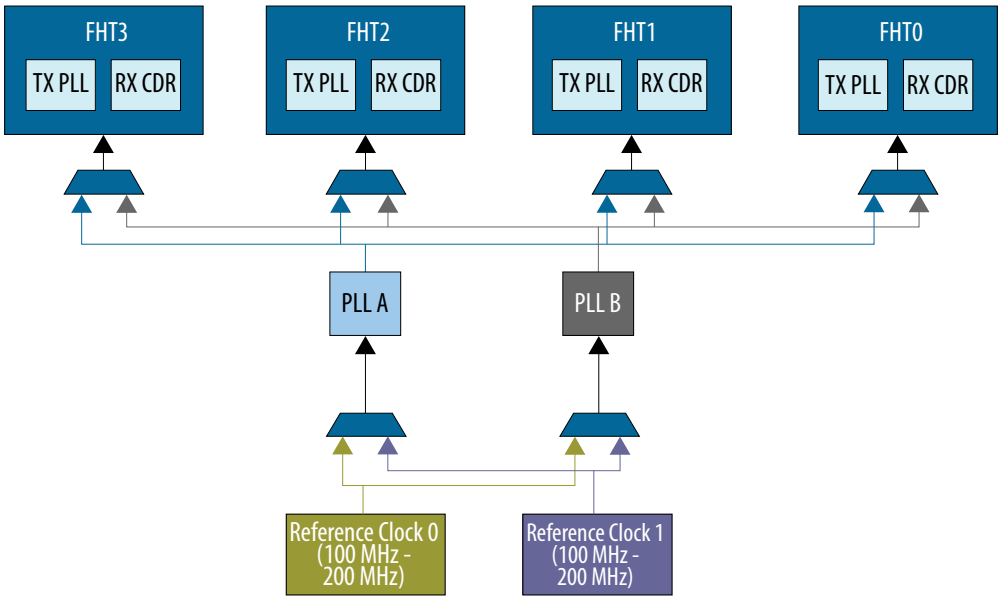


Table 21. FHT Reference Clocks

FHT Reference Clocks	Direction	Accessible FHT PMA	Accessible to System PLLs?
`refclk[0]`	Input	**FHT0**, **FHT1**, **FHT2**, **FHT3**	No
`refclk[1]`	Input	**FHT0**, **FHT1**, **FHT2**, **FHT3**	No

2.4.1.2. FGT and System PLL Reference Clock Network

There are ten reference clocks for FGT PMAs. Eight of the FGT reference clocks (`refclk[0]-refclk[7]`) can be configured as input ports. The remaining two FGT reference clocks are bidirectional. The FGT reference clock frequency range is 25-380 MHz (25-100 MHz for HDMI only).

`refclk[0]-refclk[7]` can also be shared as reference clocks for system PLLs. Refer to *System PLL* for details. The system PLL reference clock frequency range is 100-380 MHz.

There are three FGT and system PLL reference clock types.

- Global reference clocks are accessible by four FGT quads.
- Regional reference clocks are accessible by two quads.
- Local reference clocks are accessible by one quad.

Global and regional reference clocks are also accessible by system PLLs. See the following table for details.

Any hard IP that spans FGT quads must use a reference clock that is accessible by all quads. For example, PCIe x16 can only use `refclk[2]`, `refclk[3]`, `refclk[4]`, and `refclk[5]`.

Figure 50. FGT and System PLL Reference Clock Network

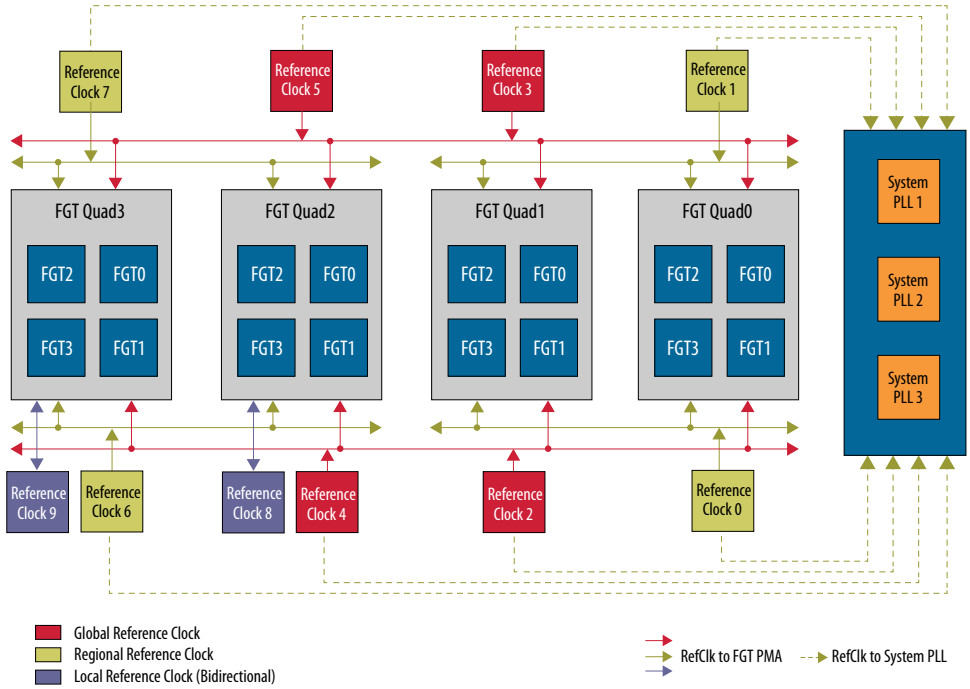


Table 22. FGT and System PLL Reference Clocks

FGT and System PLL Reference Clocks	Type	Direction	Accessible to FGT PMAs?	Accessible FGT Quads	Accessible to System PLLs?	Accessible System PLL
refclk[0]	Regional	Input	Yes	**Quad0**, **Quad1**	Yes	System PLL 1, 2, and 3
refclk[1]	Regional	Input	Yes	**Quad0**, **Quad1**	Yes	System PLL 1, 2, and 3
refclk[2]	Global	Input	Yes	**Quad0**, **Quad1**, **Quad2**, **Quad3**	Yes	System PLL 1, 2, and 3
refclk[3]	Global	Input	Yes	**Quad0**, **Quad1**, **Quad2**, **Quad3**	Yes	System PLL 1, 2, and 3
refclk[4]	Global	Input	Yes	**Quad0**, **Quad1**, **Quad2**, **Quad3**	Yes	System PLL 1, 2, and 3
refclk[5]	Global	Input	Yes	**Quad0**, **Quad1**, **Quad2**, **Quad3**	Yes	System PLL 1, 2, and 3
refclk[6]	Regional	Input	Yes	**Quad2**, **Quad3**	Yes	System PLL 1, 2, and 3
refclk[7]	Regional	Input	Yes	**Quad2**, **Quad3**	Yes	System PLL 1, 2, and 3
refclk[8]	Local	Input or output(14)	Yes	**Quad2**	No	N/A
refclk[9]	Local	Input or output(14)	Yes	**Quad3**	No	N/A

Related Information

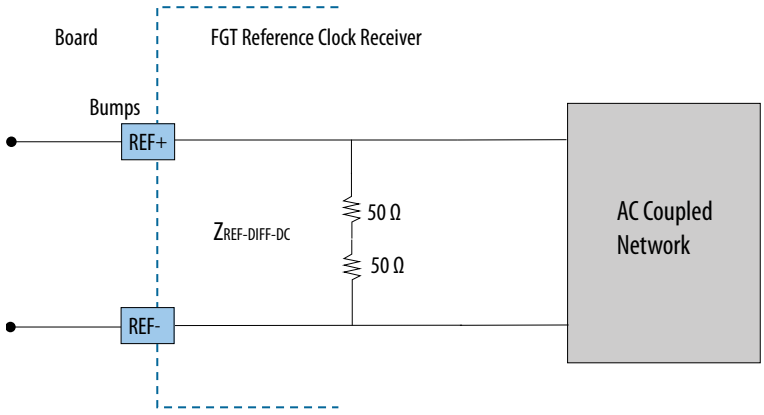
- System PLL on page 73
- F-Tile CPRI PHY Intel FPGA IP User Guide

2.4.1.2.1. FGT Reference Clock Receiver Analog Front End

A simplified FGT reference clock receiver analog front end is shown in the following figure.

(14) When configured as an output, output pins provide an RX recovered clock from one of the four FGT PMAs (from an accessible quad). You can manually select this FGT PMA at run time.
- This RX recovered clock is valid after the respective PMA achieves lock-to-data (LTD).
- The primary use case of this configuration is the CPRI protocol. Refer to the *F-Tile CPRI PHY Intel FPGA IP User Guide* for the supported recovered clock frequencies.

Figure 51. Simplified FGT Reference Clock RX Analog Front End



When FGT clock input is AC-coupled on board, no external termination or DC biasing is needed. If DC-Coupled on board, external biasing is not required unless a signaling standard other than differential 100 ohm termination is required.

Note: The on-chip internal termination registers are active once the device is powered-on, whether the device is configured or not. The on-chip internal termination registers are not active if the device is powered-off.

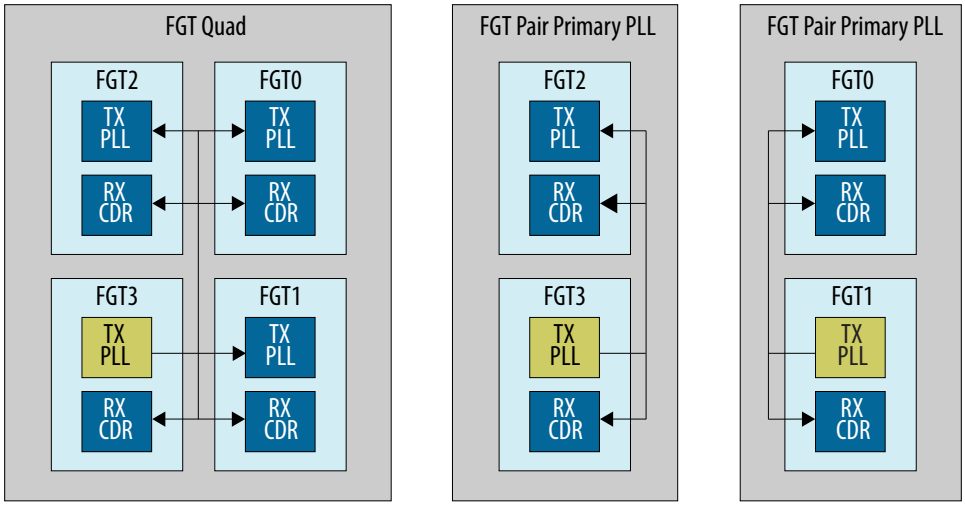
2.4.1.3. FGT Primary PLL Configuration

A primary PLL configuration is when the TX PLL of one lane is in fractional mode and acts as the reference clock source for the local CDR and TX PLL and RX CDR blocks of other lanes (configured in integer mode) within the quad. There are two different primary PLL configurations: quad and pair. These two configurations are the only supported lane combinations for a primary PLL configuration.

In a quad configuration, **FGT3** is always the primary. For example, in **Quad3**, **FGT3_Quad3** is the primary, and the **FGT3_Quad3** TX PLL output is the reference clock for the **FGT3_Quad3** RX, **FGT2_Quad3**, **FGT1_Quad3**, and **FGT0_Quad3** TX PLL, and RX CDR.

In a pair configuration with only two PMAs combined, **FGT3** or **FGT1** can be the primary. For example, when using **FGT3_Quad3** and **FGT2_Quad3**, **FGT3_Quad3** is the primary, and the **FGT3_Quad3** TX PLL output is the reference clock for the **FGT3_Quad3** RX, **FGT2_Quad3** TX PLL, and RX CDR. If using **FGT1_Quad3** and **FGT0_Quad3** , **FGT1_Quad3** is the primary.

Figure 52. FGT Primary PLL Configurations



2.4.2. Datapath Clock Network

F-tile supports two datapath clocking modes: the system PLL clocking mode and the PMA clocking mode.

- The system PLL clocking mode, in which the datapath is clocked by one of the three on-board system PLLs, is the recommended datapath clocking mode.
- The PMA clocking mode is also an option for PMA Direct.

See *PMA Data Rates* for supported data rates.

Figure 53. PMA Clocking Mode

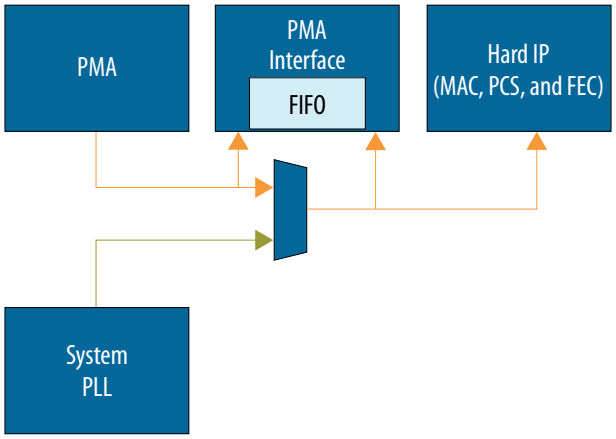
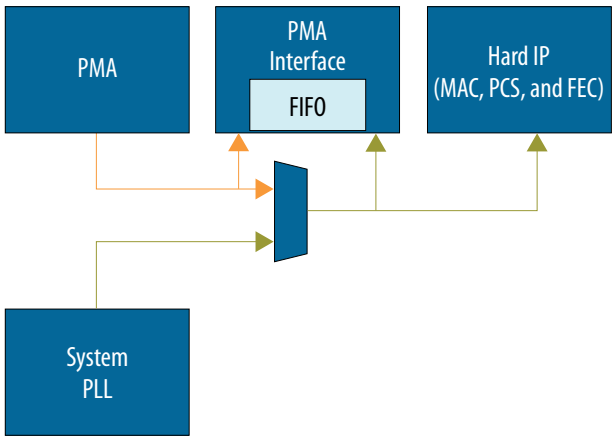


Figure 54. System PLL Clocking Mode



Related Information

PMA Data Rates on page 14

2.4.3. System PLL

F-tile has three on-board system PLLs. These system PLLs are the primary clock source for hard IP (MAC, PCS, and FEC) and EMIB crossing. This means that, when you use the system PLL clocking mode, the blocks are not clocked by the PMA clock and do not depend on a clock coming from the FPGA core. Each system PLL only generates the clock associated with one frequency interface. For example, you need two system PLLs to run one interface at 1 GHz and one interface at 500 MHz. Using a system PLL allows you to use every lane independently without a lane clock change affecting a neighboring lane.

Each system PLL can use any one of eight FGT reference clocks. System PLLs can share a reference clock or have different reference clocks. Each interface (hard IP) can choose which system PLL it uses, but, once chosen, it is fixed, not reconfigurable using dynamic reconfiguration. If PMA Direct PHY IP uses the system PLL clocking mode, PMA Direct is a data valid type interface.

With three system PLLs, you can use, for example, one system PLL for PCIe and two for Ethernet and other protocols. However, there are other use cases, and you can use all three for various interfaces within the Ethernet and PMA Direct digital blocks. Because there are only three system PLLs, multiple hard IPs with different line rates may have to share a system PLL. When multiple hard IPs share a system PLL, the hard IP with highest line rate determines the system PLL frequency, and the hard IPs with the lower line rates must be overclocked. The exact cadence is based on the clock; see *Datapath Clock Cadences* for details.

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

- The system PLL is native for the 50GbE datapath interface (the highest line rate of all four interfaces).
- The three lower line rate datapath interfaces are overclocked and need custom cadence.

Table 23. Example of a Single System PLL Shared Between Interfaces

Design	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
50GbE	53.125	64	830.08	830.08	415.04	Same as the PMA clock frequency
25GbE	25.78125	32	805.67	830.08	415.04	Over-clocked to the PMA clock frequency
24G CPRI	24.33024	32	760.32	830.08	415.04	Over-clocked to the PMA clock frequency
9.8G CPRI	9.8304	20	491.52	830.08	415.04	Over-clocked to the PMA clock frequency

Use the *F-Tile Clocking Tool* to visualize how IP and tile settings impact the datapath clocking mode. Start by reading the tool's **Introduction** tab.

Related Information

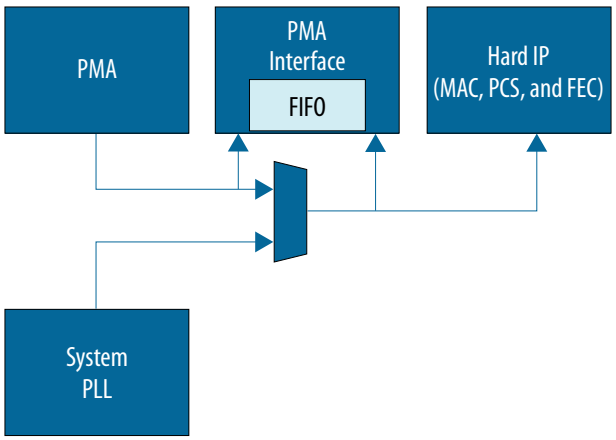
Datapath Clock Cadences on page 74

2.4.4. 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.

Figure 55. Standard Cadence and Custom Cadence



See *PMA Data Rates* for supported data rates.

Table 24. Supported Datapath Clock Frequencies and Cadences by Datapath Clocking Mode

Datapath Clocking Mode	Configuration	Datapath Clock Frequency	Cadence
PMA clocking mode (maximum 906.25 MHz)	PMA Direct	Datapath clock frequency = PMA clock frequency PMA clock frequency = line rate ÷ PMA width	Use the standard cadence on the TX and RX (data is valid at every clock edge).(15)
System PLL clocking mode (maximum 1 GHz)	PMA Direct	Use Case A: Chip-to-chip applications where F-tile 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.(16), (17)
		Use Case B: Applications where F-tile 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(18) ÷ 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; for 25GbE-1, use ≥ 805.6640625 MHz; and, for 50GbE-1, use ≥ 830.078125 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.(19)

(15) 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).

(16) 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). At 32 Gbps, only the standard cadence can be used because the system PLL reaches a maximum frequency of 1 GHz (it cannot tolerate any difference in the frequencies; the frequency delta must be 0 ppm).

(17) 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. It does not support 32.0 Gbps because the system PLL clock must run at speeds greater than 1 GHz to incorporate a frequency delta of greater than 0 ppm.

(18) 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)

(19) 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.

One 25 Gbps PMA Direct PHY IP Port Using the PMA Clocking Mode Example

- All blocks between the PMA interface and core FIFO interface run on the PMA clock.
- On the transmitter, the PMA FIFO interface is clocked by the TX PMA clock on both sides.
- On the receiver, the PMA FIFO interface 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.

Figure 56. One 25 Gbps PMA Direct PHY IP Port Using the PMA Clocking Mode Example

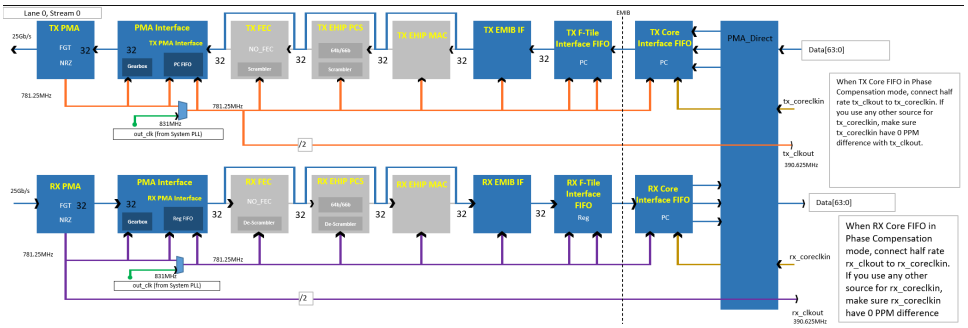
This F-Tile Clocking Tool screenshot shows one 25 Gbps PMA Direct PHY IP port using the PMA clocking mode.

Protocol	Mode	Modulation	PMA	FEC Type	Clocking Scheme		Data Rate (per lane) (Gbps)	Double Width	PMA Width	Core Interface FIFO	Operational Mode	TX/RX XCVR FIFO
PMA Direct	PMA	NRZ	FGT	NO_FEC	Transceiver Clock		25	DW	32	PC	Duplex	PC/Reg

OK: Options are Compatible

Messages & Recommendations:

1. You have to instantiate 'F-tile PMA/FEC Direct PHY IP' in Quartus to implement above selected configuration
2. This IP (configuration) must use Standard Cadence for Core Interface clocking. Standard cadence is explained in Section x.y of ABCD UG



25 Gbps Ethernet Without FEC Port Using the Overclocked System PLL Clocking Mode Example

- All blocks between the PMA interface and core FIFO interface run on the system PLL clock.
- On the transmitter, the PMA FIFO interface performs a clock transfer from the system PLL domain to the TX PMA clock domain.
- On the receiver, the PMA FIFO interface performs a clock transfer from the RX recovered clock domain to the system PLL domain. Refer to *F-Tile Ethernet Intel FPGA Hard IP User Guide* for how to clock the core interface.
- Because the system PLL clock frequency is faster than the PMA clock frequency, datapath clocking is overclocked. Therefore, you must use custom cadence.

Figure 57. 25 Gbps Ethernet Without FEC Port Using the Overclocked System PLL Clocking Mode Example

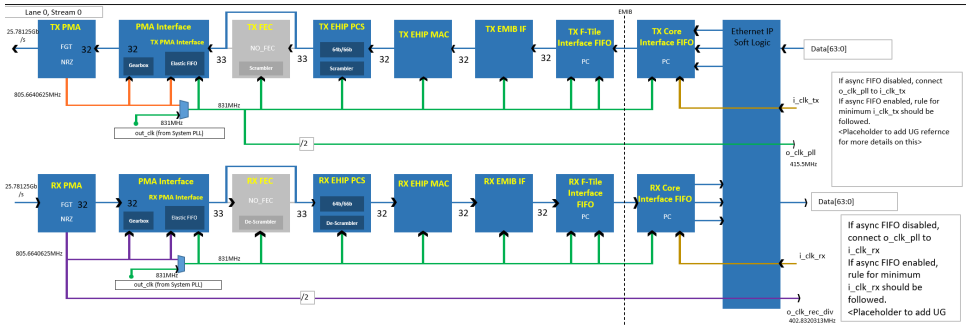
This F-Tile Clocking Tool screenshot shows one 25 Gbps Ethernet without FEC port using the overclocked system PLL clocking mode.

Protocol	Mode	Modulation	PMA	FEC Type	Clocking Scheme	System PLL Output Freq (MHz)
25GE-1	MAC_PCS	NRZ	FGT	NO_FEC	Sys PLL	831

OK: Options are Compatible

Messages & Recommendations:

1. You have to instantiate 'F-tile Ethernet Hard IP for Ethernet' in Quartus to implement above selected configuration
2. This IP (configuration) must use Custom Cadence for Core Interface clocking. To make use of Standard Cadence, System PLL output frequency need to be equal to Transceiver clock frequency (Line rate / PMA Width) at 0 PPM frequency delta. Custom cadence for Core Interface clocking is explained in Section x.y of ABCD UG.



Related Information

- PMA Data Rates on page 14
- Clock Rules and Restrictions on page 49
- F-Tile Clocking Tool
- F-Tile Ethernet Intel FPGA Hard IP User Guide

3. Implementing the F-Tile PMA/FEC Direct PHY Intel FPGA IP

The following chapters describe implementation of the Intel Agilex 7 F-tile physical (PHY) layer IP, PLLs, and clock networks. Refer to these chapters for implementation details of IP instantiation, connection, simulation, and tile placement for Intel Agilex 7 F-tile designs.

Implementation of F-tile PMA/FEC PHY designs involves instantiation and connection of the following required and optional Intel FPGA IP that is available in the Intel Quartus Prime IP catalog:

- F-Tile PMA/FEC Direct PHY Intel FPGA IP (Required)
- F-Tile Reference and System PLL Clocks Intel FPGA IP (Required)

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

- Implementing the F-Tile PMA/FEC Direct PHY Intel FPGA IP on page 78—describes function, parameters, and ports, bit mapping, core clocking, reset, and bonding of the IP.
- Implementing the F-Tile Reference and System PLL Clocks Intel FPGA IP on page 184—describes the function, parameters, and ports of the IP.
- F-tile PMA/FEC Direct PHY Design Implementation on page 199—describes instantiation, connection, simulation, and tile interface planning using an example design.

3.1. F-Tile PMA/FEC Direct PHY Intel FPGA IP Overview

The F-Tile PMA/FEC Direct PHY Intel FPGA IP enables access to the PMA Direct and FEC Direct modes via the Intel Quartus Prime IP parameter editor.

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

The F-Tile PMA/FEC Direct PHY Intel FPGA IP is for use in proprietary protocol configurations. The IP is not used as a basic building block in other Intel F-tile high-speed protocol IP, such as Ethernet, CPRI, and Interlaken. Rather, each protocol IP has its own configuration of the PMA hard block.

The following figures show the PMA direct data path and FEC direct data path with various clocking modes":

- *PMA Direct Mode with PMA Clocking*, PMA Direct Mode with PMA Clocking
- *PMA Direct Mode with System PLL Clocking*, PMA Direct Mode with System PLL Clocking
- *FEC Direct Mode with System PLL Clocking and Gearbox Enabled*, FEC Direct Mode with System PLL Clocking and Gearbox Enabled

You can use the PMA/FEC Direct PHY Intel FPGA IP to configure the datapath into PMA or FEC direct mode. If you enable the FEC mode, the FEC block is enabled as well. The top-level file that generates with the IP instance includes all the available ports for your configuration. Use these ports to connect the F-Tile PMA/FEC Direct PHY Intel FPGA IP to other IP cores in your design, such as the F-Tile Reference and System PLL Clocks Intel FPGA IP, TX and RX serial data pin IP, and the data generator and data checker IP. Refer to the block diagram in F-tile PMA/FEC Direct PHY Design IP Connections.

3.1.1. PMA Direct Supported Modes

The F-Tile PMA/FEC Direct PHY Intel FPGA IP supports the following for PMA Direct mode:

- Supports FGT with NRZ and PAM4 modulation. (20)
- Supports FHT with NRZ and PAM4 modulation
- Supports PMA clocking mode and System PLL clocking mode
- Supports Duplex, TX simplex, and RX simplex modes
- Supports TX simplex and RX simplex for both PMA Clocking and System PLL Clocking modes. Supports 8, 10, 16, 20, 32 and 64-bit PMA data widths for simplex mode.
- Supports bonding mode for NRZ and PAM4 modulation modes
- Supports configurable FIFO modes: PMA interface FIFO, F-tile interface FIFO, Core interface FIFO-elastic, phase compensation, and register modes
- Supports configuring multiple hard IP block PMA and FEC instances within one IP instance
- Supports TX and RX de-skew if the number of streams per PMA exceeds one
- Supports up to 12 Preset modes.

(20) Refer to *PMA Data Rate Ranges by Datapath Clocking Mode* for the supported data rate.

Figure 58. PMA Direct Mode with PMA Clocking

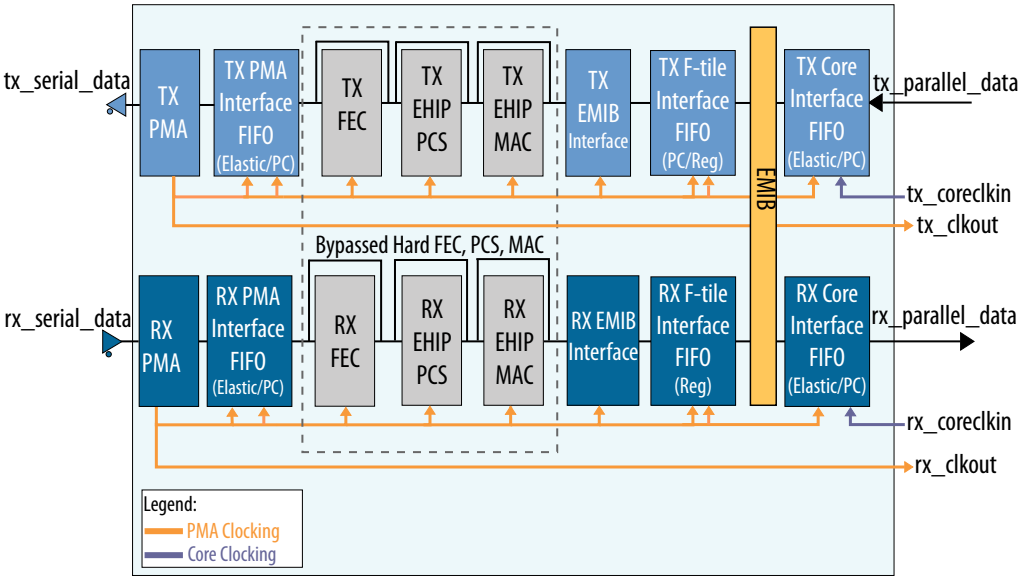


Figure 59. PMA Direct Mode with System PLL Clocking

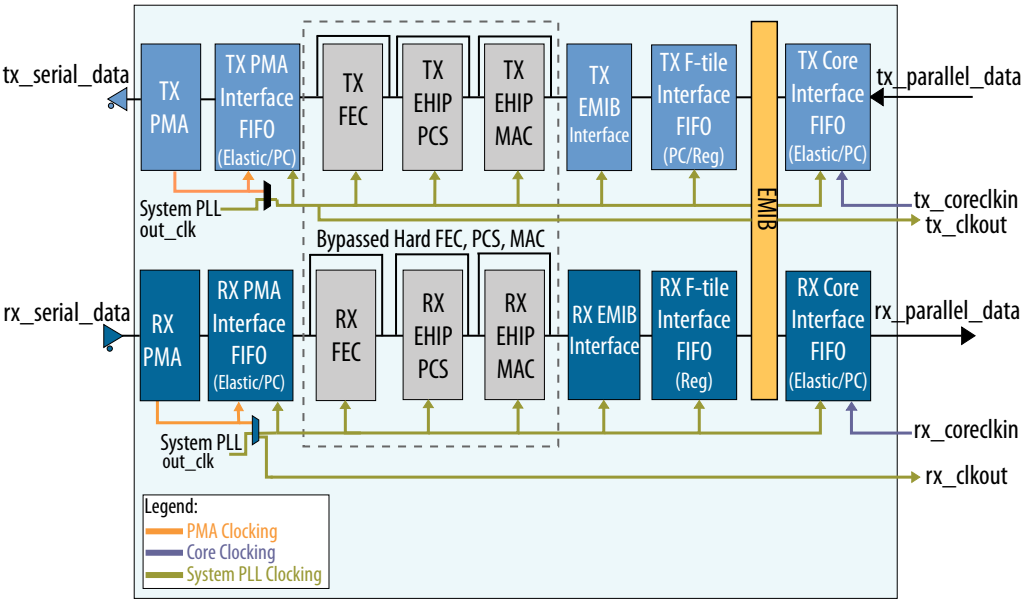


Table 25. PMA Direct Mode Support

PMA Modulation	PMA Mode	Clocking Mode	Double Width/ Single Width	PMA Interface Width	PMA Interface FIFO (TX/RX)	F-tile Interface FIFO (TX/RX[21])	Core Interface FIFO (TX/RX)
PAM4	FGT	System Clocking	DW	64, 32	Elastic/ Elastic	Phase Compensation/ Register	Phase Compensation/ Phase Compensation
			SW	32	Elastic/ Elastic	Phase Compensation/ Register	Phase Compensation/ Phase Compensation
		PMA Clocking	DW	32	Phase Compensation/ Register	TX: Register/ Phase Compensation RX: Register	TX: Phase Compensation/ Elastic| RX: Phase Compensation/ Elastic
			DW	64	Elastic/ Elastic	Phase Compensation/ Register	Phase Compensation/ Phase Compensation
	FHT	System Clocking	DW	32, 64, 128	Elastic/ Elastic	Phase Compensation/ Register	Phase Compensation/ Phase Compensation
		PMA Clocking	DW	64, 128	Elastic/ Elastic	Phase Compensation/ Register	Phase Compensation/ Phase Compensation
NRZ	FGT	System Clocking	SW	8,10, 16, 20, 32	Elastic/ Elastic	Phase Compensation/ Register	Phase Compensation/ Phase Compensation
			DW	8,10, 16, 20, 32	Elastic/ Elastic	Phase Compensation/ Register	Phase Compensation/ Phase Compensation
		PMA Clocking	SW	10, 16	Phase Compensation/ Register	Register/ Register	Phase Compensation/ Phase Compensation
				10,20, 32	Phase Compensation/ Register	Register/ Register	Elastic/ Elastic
				20, 32	Phase Compensation/	TX: Register/	Phase Compensation/

continued...

(21) In PMA direct mode, RX F-tile interface FIFO is always set to Register mode. You cannot configure this in the PMA/FEC Direct PHY Intel FPGA IP.

PMA Modulation	PMA Mode	Clocking Mode	Double Width/ Single Width	PMA Interface Width	PMA Interface FIFO (TX/RX)	F-tile Interface FIFO (TX/RX[21])	Core Interface FIFO (TX/RX)
					Register	Phase Compensation| RX: Register	Phase Compensation
			DW	20, 32	Phase Compensation/ Register	TX: Register/ Phase Compensation| RX: Register	Phase Compensation/ Phase Compensation
				20, 32	Phase Compensation/ Register	Register/ Register	Elastic/ Elastic
	FHT	System Clocking	DW	32, 64	Elastic/ Elastic	Phase Compensation/ Register	Phase Compensation/ Phase Compensation
			SW	32	Elastic/ Elastic	Phase Compensation/ Register	Phase Compensation/ Phase Compensation
		PMA Clocking	SW / DW	32	Phase Compensation/ Register	TX: Register/ Phase Compensation| RX: Register	Phase Compensation/ Phase Compensation
			DW	64	Elastic/ Elastic	Phase Compensation/ Register	Phase Compensation/ Phase Compensation

3.1.2. FEC Direct Supported Modes

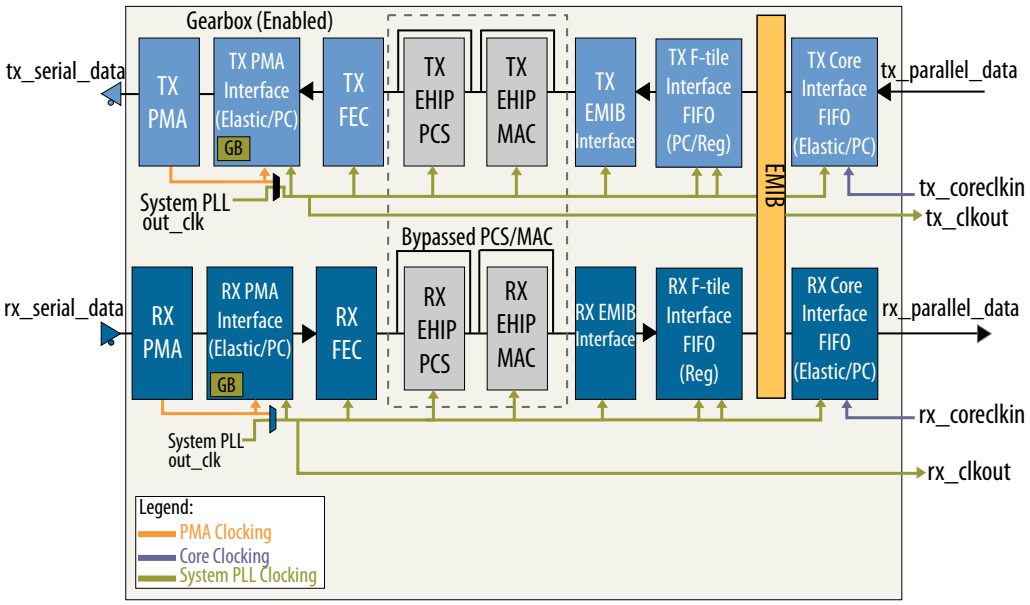
The F-Tile PMA/FEC Direct PHY Intel FPGA IP supports the following for FEC Direct mode:

- IEEE 802.3 RS(528, 514) (CL 91, KR)
- IEEE 802.3 RS(544,514) (CL 134, KP)
- Ethernet Technology Consortium LL RS(272, 258)
- Supports 25-400G KP/KR/LL FEC
- Supports only the System PLL clocking mode
- Supports only the Duplex operation mode
- Supports the gearbox feature

(21) In PMA direct mode, RX F-tile interface FIFO is always set to Register mode. You cannot configure this in the PMA/FEC Direct PHY Intel FPGA IP.

You can enable FEC Direct mode in the IP parameter editor by turning on the **Enable RS-FEC** option, as RS-FEC (Reed Solomon Forward Error Correction) Options on page 104 describes. FEC direct modes (KP,KR,LL) with FEC specifications are topology-dependent to achieve different BER. FEC data to and from the PCS is 33b. On the PMA interface side, FEC data from and to the PMA interface is 40b wide.

Figure 60. FEC Direct Mode with System PLL Clocking and Gearbox Enabled



Enabling Gearbox in RS-FEC Mode

For designs that include FEC, gearbox enables automatically. The gearbox options are 32:40, 64:80, and 128:160. Firecode FEC supports the 32:33 gearbox ratio. For designs that include a PCS (Physical Coding Sublayer) only, the only option is the 32:33 gearbox ratio.

Table 26. FEC Direct IP Configuration Mode Support

Mod Type	PMA Type	FEC Mode	Clock Mode	Width	PMA Width	PMA Interface FIFO (Tx/Rx)	F-tile Interface FIFO (Tx/Rx)	Core Interface FIFO (Tx/Rx)
NRZ	FGT	RS(528, 514), RS(272,258), RS(544, 514)	Sys PLL	DW	32	Elastic/ Elastic	Phase Compensation/ Phase Compensation	Phase Compensation/ Phase Compensation
	FHT	RS(528, 514), RS(272,258), RS(544, 514)	Sys PLL	DW	32,64	Elastic/ Elastic	Phase Compensation/ Phase Compensation	Phase Compensation/ Phase Compensation
PAM4	FGT	RS(528, 514), RS(272,258),	Sys PLL	DW	32,64	Elastic/ Elastic	Phase Compensation/	Phase Compensation/

continued...

Mod Type	PMA Type	FEC Mode	Clock Mode	Width	PMA Width	PMA Interface FIFO (Tx/Rx)	F-tile Interface FIFO (Tx/Rx)	Core Interface FIFO (Tx/Rx)
		RS(544, 514)					Phase Compensation	Phase Compensation
	FHT	RS(528, 514), RS(272,258), RS(544, 514)	Sys PLL	DW	32, 64, 128	Elastic/ Elastic	Phase Compensation/ Phase Compensation	Phase Compensation/ Phase Compensation

3.1.3. Unsupported PMA/FEC Modes

The F-Tile 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 FHT and FEC direct mode.
- Gearbox feature is disabled for PMA Direct mode.
- No support for mixtures of PMA modulation modes within the same PMA . That is, a TX-only and RX-only channel cannot be a mixture of NRZ and PAM4 modulation modes.
- Does not support the reset controller function directly. Rather, reset ports that you expose by means of this IP core route to the tile-wide soft reset controller (SRC).
- Parameter editor does not support analog parameter settings, such as termination, coupling, or transmitter PMA equalizer parameters. Rather, you must configure analog pin settings, such as termination and coupling, by use of `.qsf` assignments. This method allows you to reuse the same IP instance for different applications, with various pin analog settings, in different `.qsf` files.
- No support for enabling a rate-matching soft FIFO between the user FPGA core logic and the IP for pacing the data valid signal inside the IP.

3.2. Designing with F-Tile PMA/FEC Direct PHY Intel FPGA IP

The F-Tile 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 F-tile PMA block features for both FGT and FHT.

To customize and instantiate the IP for your protocol implementation, you specify parameter values for the F-Tile 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.

You can use the F-Tile PMA/FEC Direct PHY Intel FPGA IP in your design if a custom PCS or MAC block is created using your own logic, rather than using the Intel FPGA PCS or MAC block.

The F-Tile PMA/FEC Direct PHY Intel FPGA IP allows you to configure the F-tile FGT and FHT to support PMA and FEC direct modes with the following:

- Predefined preset parameters for IP
- Datapath Clocking mode, PMA type, PMA modulation type, PMA data rate
- TX datapath and RX Datapath options settings (FIFO modes, TX PLL, RX CDR)
- RS-FEC Modes and options
- Datapath Avalon Memory Mapped Interface, PMA Avalon Memory Mapped Interface

3.2.1. Preset IP Parameter Settings

The IP parameter editor provides preset settings for the F-Tile PMA/FEC Direct PHY Intel FPGA IP. You can specify the preset settings as a starting point for your design.

To apply preset parameters, double-click the preset name, and click **Apply**. For example, selecting the **FGT_NRZ_50G_2_PMA_Lanes_Custom_Cadence_ED** preset enables all parameters and ports that the PMA Direct mode requires, with two FGT PMA operating at 25.78125Gbps.

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

Table 27. F-Tile PMA/FEC Direct PHY Intel FPGA IP Available Parameter Presets

PMA / FEC Direct Mode Preset	Link	Fracture Type	PMA Data Rates
FGT_NRZ_128GFC_4_PMA_Lanes_RSFEC_528_514	128 Gbps FEC Direct FGT NRZ Link	4 st_x1 fractures	4 PMA Lanes of 25.78125 Gbps
FGT_NRZ_150G_6_PMA_Lanes_System_PLL	150 Gbps PMA Direct FGT NRZ Link	6 st_x1 fractures	6 PMA Lanes of 25.78125 Gbps
FGT_NRZ_200G_8_PMA_Lanes_RSFEC_528_514	200 Gbps FEC Direct FGT link	8 st_x1 fractures	8 PMA lanes of 25.78125 Gbps

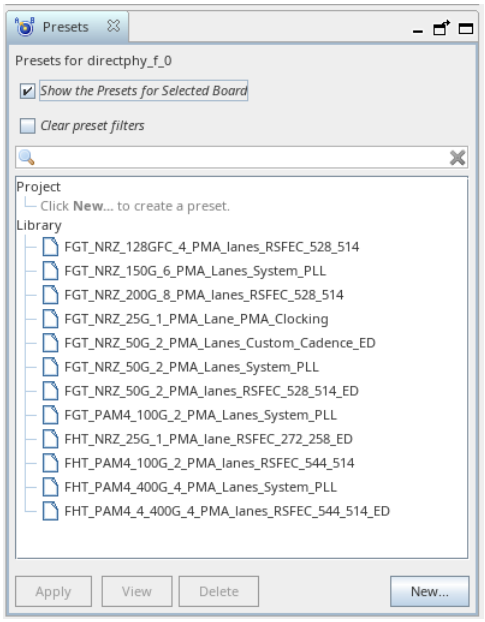
continued...

PMA / FEC Direct Mode Preset	Link	Fracture Type	PMA Data Rates
FGT_NRZ_25G_1_PMA_Lane_PMA_Clocking	25 Gbps PMA Direct FGT NRZ Link	1 st_x1 fracture	1 PMA Lane of 25.78125 Gbps
FGT_NRZ_50G_2_PMA_Lanes_System_PLL	50 Gbps PMA Direct FGT NRZ Link	2 st_x1 fractures	2 PMA Lanes of 25.78125 Gbps
FGT_PAM4_100G_2_PMA_Lanes_System_PLL	100 Gbps PMA Direct FGT PAM4 Link	2 st_x2 fractures	2 PMA Lanes of 53.125 Gbps
FHT_PAM4_100G_2_PMA_Lanes_RSFEC_544_514	100 Gbps FEC Direct FHT PAM4 Link	2 st_x2 fractures	2 PMA Lanes of 53.125 Gbps
FHT_PAM4_400G_4_PMA_Lanes_System_PLL	400Gbps PMA Direct FHT PAM4 Link	4 st_x4 fractures	4 PMA Lanes of 106.25 Gbps
FHT_PAM4_400G_4_PMA_lanes_RSFEC_544_514_ED[22]	400 Gbps FEC Direct FHT PAM4 link	4 st_x4 fractures	4 PMA Lanes of 106.25 Gbps
FGT_NRZ_50G_2_PMA_lanes_RSFEC_528_514_ED[22]	50 Gbps FEC Direct FGT NRZ Link	2 st_x1 fractures	2 PMA Lanes of 25.78125 Gbps
FHT_NRZ_25G_1_PMA_lane_RSFEC_272_258_ED[22]	25 Gbps FEC Direct FGT NRZ Link	1 st_x1 Fracture	1 PMA Lane for 25.78125 Gbps
FGT_NRZ_50G_2_PMA_Lanes_Custom_Cadence_ED[22]	50 Gbps PMA Direct FGT NRZ link	2 st_x1 Fracture	2 PMA Lanes for 25.78125 Gbps

Note: Refer to F-Tile Building Blocks on page 8 for fracture type descriptions.

[22] Support for example design generation.

Figure 61. Available Parameter Presets In Parameter Editor



3.3. Configuring the IP

Use the F-Tile 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 7 F-tile device, such as **AGIB027R31B1E2V**.
2. Click **Tools ➤ IP Catalog**, type pma in the search field, and select **F-Tile 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 F-Tile 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 for FGT and FHT
   - FEC Direct mode for FGT and FHT
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 F-Tile PMA/FEC Direct PHY Intel FPGA IP to other IP cores in your design, such as the F-Tile Reference and System PLL Clocks Intel FPGA IP, TX and RX serial data pins, and the data generator and data checker IP.

The F-Tile PMA/FEC Direct PHY Intel FPGA IP supports only the following simulators for this release:

- VCS*
- ModelSim* SE
- QuestaSim*
- Xcelium*

Related Information

- F-tile PMA/FEC Direct PHY Design Implementation on page 199
- Connecting the F-tile PMA/FEC Direct PHY Design IP on page 212

3.3.1. General and Common Datapath Options

You can customize your instance of the F-Tile PMA/FEC Direct PHY Intel FPGA IP by specifying parameter values to suit your application. The parameters are organized in the following sections for each functional block and feature:

- **General and Common Datapath** options
- TX Datapath Options on page 92
- RX Datapath Options on page 97
- RS-FEC (Reed Solomon Forward Error Correction) Options on page 104
- Avalon Memory Mapped Interface Options on page 106

Figure 62. F-Tile PMA/FEC Direct PHY Intel FPGA IP Parameter Editor

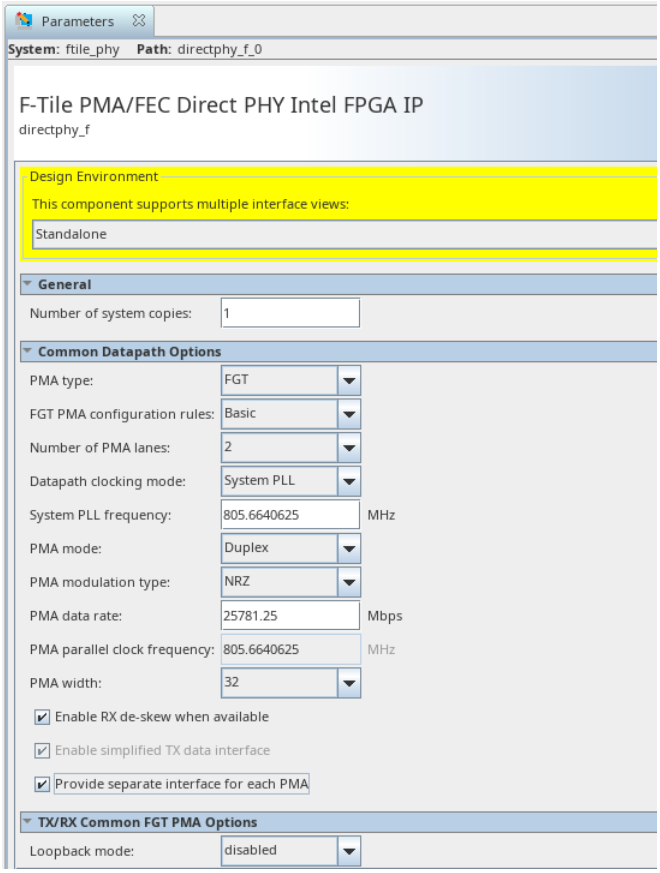


Table 28. General and Common Datapath Options

Parameter	Values	Description
General		
Number of system copies	1-16	Specifies the total number of independent system copies. For example, you can create multiple copies of the same PMA/FEC mode configuration by setting this to value of 2 (or more) and with the same PMA configuration. This is duplicated to create 2 PMA lanes of the same configuration for the given example. Default value is **1**.
Common Datapath Options		
PMA type	**FGT**,**FHT**	Specifies the type of PMA used. **FGT** PMA can operate up to 58.125Gbps PMA. **FHT** PMA can operate up to 116Gbps. Default is **FGT**.
FGT PMA configuration rules	**Basic**,**OTN**, **CPRI**, **GPON**, **SDI**, **SONET**, **SATA**, **USB**	Selects the protocol configuration rules for the FGT 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.
Number of PMA lanes	**1**, **2**, **4**, **6**, **8**, **12**, **16**	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 System bonding. Values 6 and 12 supported only for PMA Direct mode. Default value is **1**.

continued...

Parameter	Values	Description
Datapath clocking mode	**PMA** **System PLL**	Specifies whether the PMA parallel clock or System PLL is used to clock the TX/RX datapath. Use of System PLL is required when **Enable RS-FEC** is on or dynamic reconfiguration is intended. Default value is **System PLL**.
System PLL frequency	31.25 to 1000	Specifies the System PLL clock frequency (MHz) and applicable if datapath clocking mode is selected as System PLL. Default value is **830.08**[23].
PMA mode	**Duplex**, **TX Simplex**, **RX Simplex**	Specifies the PMA operation mode. TX simplex and RX simplex can operate at independent rates. Default value is **Duplex**.
PMA modulation type	**PAM4**,**NRZ**	Specifies the modulation type used for serial data. Default value is **PAM4**.
PMA data rate	25781.25	Specifies the PMA data rate in units of Mbps (megabits per second). Default value is **25781.25**.
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**. The PMA parallel clock frequency applies to both the **Word Clock** and **Bond Clock**.
PMA width	**8**, **10**, **16**, **20**, **32**, **64**, **128**	Specifies the PMA data width. 128-bit is only supported for **FHT**. The PMA data width specifies the total number of PMA bond streams (PMA bonding). For example: • data width 8, 10, 16, 20 and 32-bit with 1 PMA stream (no PMA bonding) • 64-bit with 2 PMA bonded streams • 128-bit with 4 PMA bonded streams Default value is **64**.
Enable RX de-skew when available	On/Off	Enables the RX de-skew feature. This feature is only available in: • PMA Direct mode single lane when using PAM4 with PMA width of 64 or 128. Multi-lane when using PAM4 with PMA width of 64 or 128 • FEC Direct mode 64- or 128-bit (both **NRZ** and **PAM4** enabled). Only supported in multi-lanes (Number of PMA lanes is greater than 1) Default value is **On**.
Enable simplified TX data interface	On/Off	Enables simplified data and control interface between the FPGA and PMA for SATA and USB protocol modes. When the **FGT PMA configuration rules** parameter is set to SATA or USB, you can enable this option to control the `fgt_tx_pma_elecidle` port. Refer to TX PMA Control Signals for more information.
Provide separate interface for each PMA	On/Off	When On, the PMA/FEC Direct PHY IP presents separate data and clock interfaces for each PMA lane, rather than a wide bus. Default value is **Off**. Refer to Signal and Port Reference on page 110 for a list of signals that are not impacted by this feature. *Note:* When the **Enable RS-FEC** option is on, a separate interface is not available for each PMA by use of the **Provide separate interface for each PMA** option.

(23) Refer to Guidelines for F-Tile Reference and System PLL Clocks Intel FPGA IP Usage on page 195 for reference clock and system PLL usage.

3.3.1.1. FGT PMA Configuration Rules for SATA and USB mode

Note: For the SATA and USB protocol modes, support for the following features is preliminary in both simulation and hardware:

- PMA electrical idle feature
- Squelch detect feature
- Signal detect feature

The final support for these features is planned in a future version of the Intel Quartus Prime Pro Edition software.

You can implement the SATA or USB protocol modes with the F-Tile PMA/FEC Direct PHY Intel FPGA IP by following the steps shown below:

1. In the **General and Common Datapath Options**:
   a. Select **SATA** or **USB** for the **FGT PMA configuration rules** setting.
   b. Enable the **Simplified TX data interface** setting.
      - This step turns on the `fgt_tx_pma_electricle` idle port which is part of the protocol.
2. In the **TX Datapath Options**:
   a. Enable the **Enable Spread Spectrum clocking** setting.
3. In the **RX Datapath Options**:

   For SATA:

   a. Enable the **Enable SATA squelch detection** setting.
   b. Enable the **Enable fgt_rx_signal_detect_lfps port** setting for low frequency periodic signaling.
   c. Enable the **Enable fgt_rx_signal_detect port** setting for out of band signaling.

   For USB:

   a. Enable the **Enable fgt_rx_signal_detect_lfps port** setting for low frequency periodic signaling.

Related Information

- TX FGT Datapath Parameters on page 92
- RX FGT Datapath Parameters on page 98

3.3.2. TX Datapath Options

Figure 63. TX FGT PMA Parameters in Parameter Editor

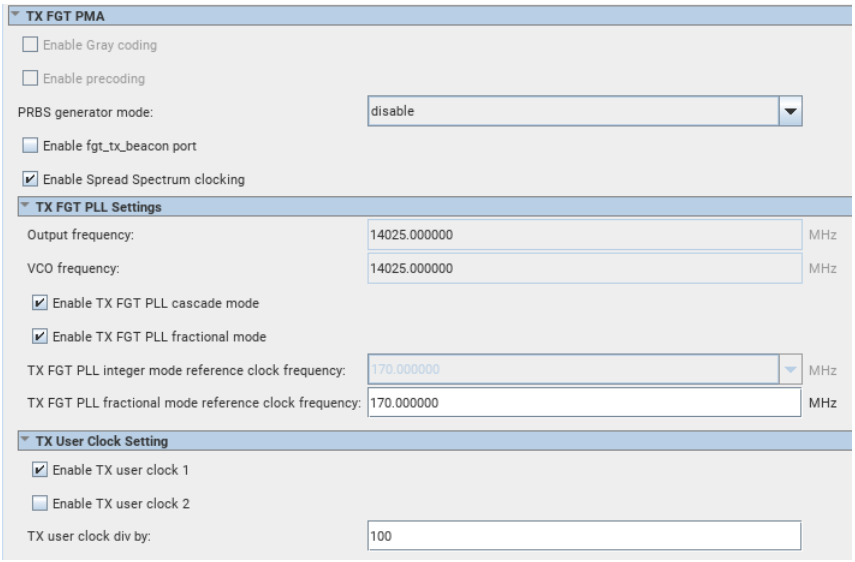


Table 29. TX FGT Datapath Parameters

Parameter	Values	Description
TX FGT PMA Parameters		
Enable Gray coding	**On/Off**	Enables Gray coding. Applicable to PAM4 encoding only. When Off, TX sends gray code set to 0xB4. When On, TX sends gray code set to 0x6C. Must be **Off** for normal operation, or when in internal or external loopback mode. Default value is **Off**.
Enable precoding	**On/Off**	Enables pre-coding. Applicable to PAM4 encoding only. Default value is **Off**.
PRBS generator mode(24)	**disable**, **PRBS7**, **PRBS9**, **PRBS10**, **PRBS13**, **PRBS15**, **PRBS23**, **PRBS28**, **PRBS31**, **QPRBS13**, **PRBS13Q**, **PRBS31Q**, **SSPR**, **SSPR1**, **SSPRQ**	Enables hard PRBS generator with the PRBS polynomial selection. Default value is **disable**.
Enable fgt_tx_beacon port	**On/Off**	Enables fgt_tx_beacon port for SATA.
Enable Spread Spectrum clocking	**On/Off**	Enables spread spectrum clocking for SATA and USB protocol compliance.
TX FGT PLL Parameters		
Output frequency	N/A	Shows the calculated TX FGT PLL output frequency.

continued...

(24) The **PRBS31**, **QPRBS13**, **PRBS13Q**, **PRBS31Q**, **SSPR**, **SSPR1**, and **SSPRQ PRBS generator mode** settings are not currently supported through the IP GUI, although present in the parameter editor. Do not select any of the unsupported **PRBS generator mode** settings. Specify these settings using registers.

Parameter	Values	Description
VCO frequency	N/A	Shows the calculated TX FGT PLL VCO output frequency.
Enable TX FGT PLL cascade mode	**On/Off**	Enable cascade mode for Duplex link only. Default value is **Off**. Refer to FGT PMA Fractional Mode on page 141 for more detail.
Enable TX FGT PLL fractional mode	**On/Off**	Enables TX FGT PLL's fractional mode. Default value is **Off**. Refer to FGT PMA Fractional Mode on page 141 for more detail.
TX FGT PLL integer mode reference clock frequency	**25 to 380 MHz**	Selects the reference clock frequency (MHz) for the TX FGT PLL. Range is: • 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA(23).
TX FGT PLL fractional mode reference clock frequency	**25 to 380 MHz**	Selects the reference clock frequency (MHz) in fractional mode for the TX FGT PLL. Range is: • 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA(23).
TX User Clock Parameters		
Enable TX user clock 1	**On/Off**	Enables and disables TX user `clock1`. If the clock is not used, you can disable this to save power. Default value is **On**.
Enable TX user clock 2	**On/Off**	Enables and disables TX user `clock2`. If the clock is not used, you can disable it to save power. Default value is **Off**.
TX user clock div by	**12- 139.5**	Divider values for the TX PLL VCO output frequency. Values from 12 to 139.5 are acceptable in 0.5 increments. The same dividers are shared for both TX user clock 1 and 2. Default value is **100**.

Figure 64. TX FHT PMA Parameters in Parameter Editor

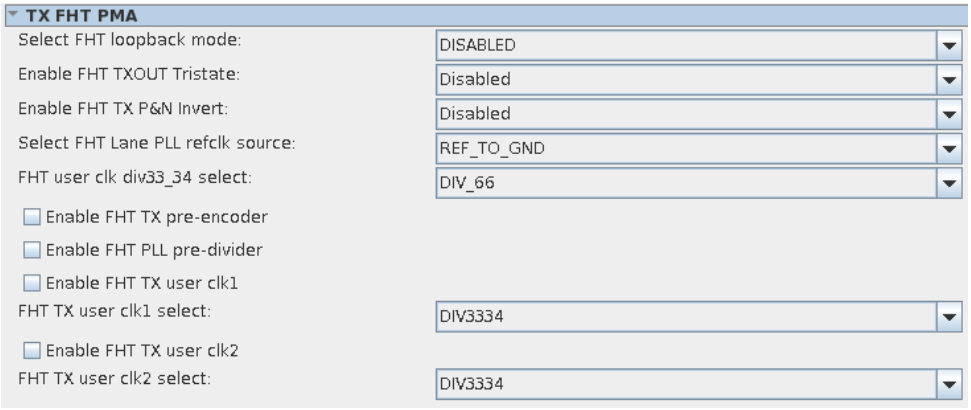


Table 30. TX FHT PMA Parameters

Parameter	Values	Description
TX FHT PMA Parameters		
Select FHT loopback mode	**PARALLEL_LOOPBACK**, **SERIAL_EXT_LOOPBACK**, **SERIAL_ANA_LOOPBACK**, **REVERSE_PARALLEL_LOOPBACK**, **WRAP_LOOPBACK**, **DISABLED**	Enables FHT loopback modes. Default is **DISABLED**.
Enable FHT TX P&N Invert	**Disabled**/**Enabled**	Enable this to invert TX P and N output. Default is **Disabled**.
Select FHT Lane PLL refclk source	**REF_TO_GND**, **CDR_PLL_CLK**, **PLL_100_MHZ**, **PLL_156_MHZ**	Selects the FHT Lane PLL refclk source. • **REF_TO_GND** disables TX lane PLL • **CDR_PLL_CLK** selects RX recovered as the source of reference clock for TX lane PLL • **PLL_100_MHZ** selects 100 MHz output from common PLL to lane PLL • **PLL_156_MHZ** selects 156.26 MHz output from common PLL to lane PLL Default value is **REF_TO_GND**.
FHT user clk div33_34 select	**DIV_33** **DIV_34** **DIV_66** **DIV_68**	Selects one of the four DIV clock output for the TX user clock. Refer to Clocking on page 134 for more details on how to use this output.
Enable FHT PLL pre-divider	**On/Off**	Enables FHT PLL pre-divider. Default value is **Off**. If disabled, pre-divider value is 1 and if enabled pre-divider value is 2. In certain configurations, where disabling this sets the lane PLL to fractional mode, you must enable this to set the lane PLL in integer mode for better performance.
Enable FHT TX pre-encoder	**On/Off**	Enables FHT TX pre-encoder. Default value is **Off**. This setting must match the link partner's RX pre-encoder setting.
Enable FHT TX user clk1	**On/Off**	Enables the FHT TX user clk1. Default value is **Off**.
FHT TX user clk1 select	**DIV3334** **DIV40**	FHT TX user clk1 select, you can select DIV3334 (one of the four DIV clocks listed in user div33_34) or DIV40 clock. Default value is **DIV3334**. Refer to Clocking on page 134.
Enable FHT TX user clk2	**On/Off**	Enables FHT TX user clk2. Default value is **Off**.
FHT TX user clk2 select	**DIV3334** **DIV40**	FHT TX user clk2 select, you can select DIV3334 or DIV40 clock. Default value is **DIV3334**. Refer to Clocking on page 134.

3.3.2.1. TX PMA interface Parameters

Figure 65. TX PMA interface Parameters

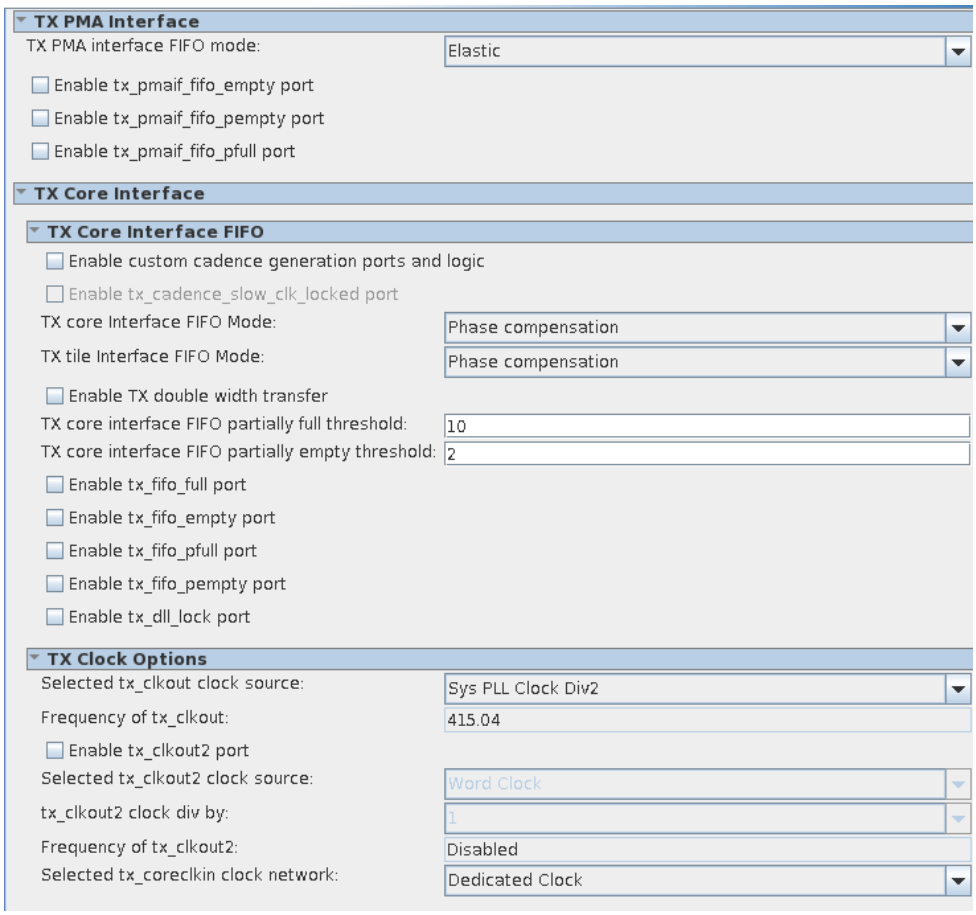


Table 31. TX PMA interface Parameters

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**.
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 (`tx_cadence`, `tx_cadence_fast_clk`, `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 on page 145 .

continued...

Parameter	Values	Description
Enable tx_cadence_slow_clk_locked port	On/Off	If `tx_cadence_slow_clk` is not directly coming from TX PLL (word clock/bond clock/user clock), but rather comes from another clock source, you must turn on the **tx_cadence_slow_clk_locked** port option in the parameter editor. `tx_cadence_slow_clk_locked` 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 FIFO is only supported for PMA Clocking mode.
TX Tile Interface FIFO mode	**Phase Compensation** **Register**	Specifies the mode for the TX Tile Interface FIFO. Default value is **Phase Compensation**.
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**.
TX core interface FIFO partially full threshold	10	Specifies the partially full threshold for the TX Core Interface FIFO. Default value is **10**.
TX core interface FIFO partially empty threshold	2	Specifies the partially empty threshold for the TX Core Interface FIFO. Default value is **2**.
Enable tx_fifo_full port	On/Off	Enables the optional `tx_fifo_full` status output port. This signal indicates when the TX core FIFO has reached the full threshold. This signal is synchronous with `tx_clkout`. Default value is **Off**.
Enable tx_fifo_empty port	On/Off	Enables the optional `tx_fifo_empty` status output port. This signal indicates when the TX core FIFO has reached the empty threshold. This signal is synchronous with `tx_clkout`. Default value is **Off**.
Enable tx_fifo_pfull port	On/Off	Enables the optional 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 `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**.
Enable tx_dll_lock port	On/Off	Enables the optional `tx_dll_lock` status output port. Monitor this signal when the core interface FIFO is in elastic mode, and then wait for the `tx_dll_lock` port to assert before asserting the write enable bit for the core interface FIFO. This signal indicates when the TX DLL is locked for data transfer. Default value is **Off**. Refer to TX and RX Parallel Data Mapping Information for Different Configurations on page 126 for the write enable bit.
TX Clock Options		
Selected tx_clkout clock source	**Word Clock** **Bond Clock** **User Clock 1** **User Clock 2** **Sys PLL Clock** **Sys PLL Clock Div2**	Specifies the `tx_clkout` output port source. Default value is **Sys PLL Clock Div2**.
Frequency of tx_clkout	Output	Displays the frequency of `tx_clkout` in MHz based on `tx_clkout` source selection.

continued...

Parameter	Values	Description
Frequency of tx_clkout2	Output	Displays the frequency of `tx_clkout2` in MHz based on `tx_clkout2` source selection and `tx_clkout2` clock divide by factor.
Enable tx_clkout2 port	On/Off	Enables the optional `tx_clkout2` output clock. Default value is **Off**.
Selected tx_clkout2 clock source	**Word Clock** **Bond Clock** **User Clock 1** **User Clock 2** **Sys PLL Clock** **Sys PLL Clock Div2**	Specifies the `tx_clkout2` output port source. Default value is **Word Clock**.
tx_clkout2 clock div by	**1**, **2**, **4**	Selects the tx_clkout2 divider setting that divides out the `tx_clkout2` output port source. Default value is **1**.
Selected tx_coreclkin clock network	**Dedicated Clock** **Global Clock**	Specifies the type of clock network to use to route the clock signal to `tx_coreclkin` port. Dedicated clock allows a higher maximum frequency between the FPGA fabric and the F-tile interface. The number of **Dedicated Clock** lines are limited. Default value is **Dedicated Clock**.

3.3.3. RX Datapath Options

Figure 66. RX Datapath Options in Parameter Editor

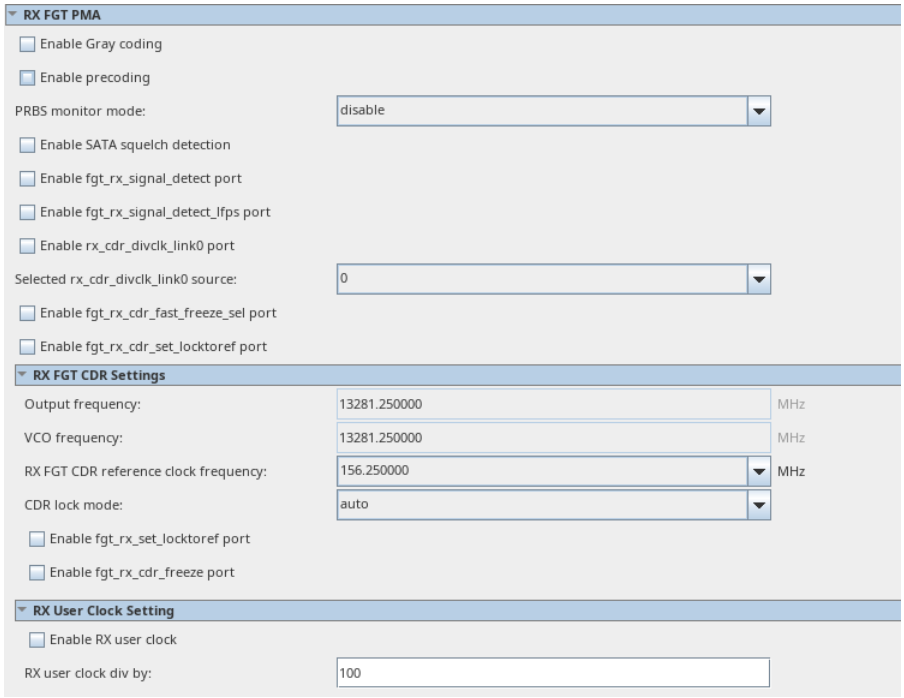


Table 32. RX FGT PMA Parameters

Parameter	Values	Description
Enable Gray coding	**On/Off**	Enables Gray coding. Applicable to PAM4 encoding only. When Off, link partner must send gray code set to 0xB4. When On, link partner must send gray code set to 0x6C. Must be Off for normal operation, or when in internal/external loopback mode). Default value is **Off**.
Enable precoding	**On/Off**	Enables precoding. Applicable to PAM4 encoding only. Default value is **Off**.
PRBS monitor mode(25)	**disable**, **PRBS7**, **PRBS9**, **PRBS10**, **PRBS13**, **PRBS15**, **PRBS23**, **PRBS28**, **PRBS31**, **QPRBS13**, **PRBS13Q**, **PRBS31Q**, **SSPR**, **SSPR1**, **SSPRQ**	Enables hard PRBS generator with the PRBS polynomial selection. Default value is **disable**.
Enable SATA squelch detection	**On/Off**	Enables squelch detection for SATA. Default value is **Off**.
Enable fgt_rx_signal_detect port	**On/Off**	Enables the `fgt_rx_signal_detect` port. This port is used for SATA protocol mode for out of band (OOB) signal detection. Default value is **Off**.
Enable fgt_rx_signal_detect_lfps port	**On/Off**	Enables the `fgt_rx_signal_detect_lfps` port. This port is used for SATA and USB protocol mode for low frequency periodic signaling (LFPS) signal detection. Default value is **Off**.
Enable rx_cdr_divclk_link0 port	**On/Off**	Enables the link port representing RX CDR clock output from RX PMA to the reference clock pin. The connection made from this port to the F-Tile Reference and System PLL Clocks Intel FPGA IP guides the Fitter to determine the physical pin. Do not use this pin itself in simulation to observe clock behavior. Observe the actual clock behavior in the related output port of the F-Tile Reference and System PLL Clocks Intel FPGA IP. The physical port is typically used for CPRI. You can connect the physical port to the physical reference clock pin 8 or 9 for configuration as RX CDR clock output. This setting is applicable for FGT PMA only. Default value is **Off**.
Selected rx_cdr_divclk_link0 source	0 to N-1, N = Number of PMA Lanes)	Determines which RX FGT PMA lane is sourcing `fgt_rx_cdr_divclk_link0`. Note that FGT PMA index used in this parameter is logical. The selected PMA lane must be physically mapped to FGT Quad 3 (with reference clock 9) or FGT Quad 2 (with reference clock 8). If **Enable rx_cdr_divclk_link0 port** is off, this parameter is ignored. Default value is **Off**.
Enable fgt_rx_cdr_fast_freeze_sel port	**On/Off**	This port is used for GPON. For GPON mode, you must enable and tie the `fgt_rx_cdr_fast_freeze_sel` signal to 1'b0. It allows `fgt_rx_cdr_freeze` control signal to propagate correctly. Default value is **Off**.

continued...

(25) The **PRBS31**, **QPRBS13**, **PRBS13Q**, **PRBS31Q**, **SSPR**, **SSPR1**, and **SSPRQ PRBS generator mode** settings are not currently supported through the IP GUI, although present in the parameter editor. Do not select any of the unsupported **PRBS generator mode** settings. Specify these settings using registers.

Parameter	Values	Description
Enable fgt_rx_cdr_set_locktoref port	**On/Off**	Primarily used for GPON. When enabled, asserting the `fgt_rx_cdr_set_locktoref` signal keeps the CDR in lock-to-reference mode. If CDR lock mode is in lock to reference, then asserting this signal keeps CDR in lock to reference mode. Deasserting this signal keeps CDR in auto mode. When switching modes you have to assert reset. In manual reference clock mode, reset controller should be switched to ignore locktodata mode through appropriate write to soft CSRs. Default value is **Off**.
RX FGT CDR Settings		
Output frequency	12890.625MHz	Specifies the non editable RX FGT CDR output frequency initial value derived from the IP configuration.
VCO frequency	12890.625MHz	Specifies the non editable RX FGT CDR VCO output frequency initial value derived from the IP configuration.
RX FGT CDR reference clock frequency	25.781250-250.000000	Selects the reference clock frequency (MHz) for CDR. Default value is **156.25**.
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 fgt_rx_set_locktoref to control the CDR lock behavior. If fgt_rx_set_locktoref is low CDR operates in auto mode, and in lock to reference mode if it is high. In manual mode, reset controller should be notified to ignore lock to data status through appropriate soft CSR write. Default value is **auto**.
Enable fgt_rx_set_locktoref port	**On/Off**	Asserting this signal keeps CDR in lock to reference mode. Deasserting this signal keeps CDR in auto mode. When switching modes, assert reset. In manual reference clock mode, switch the reset controller to ignore locktodata mode through appropriate write to soft CSRs. Default value is **Off**.
Enable fgt_rx_cdr_freeze port	**On/Off**	This port is used for GPON to freeze the CDR lock state during non-active time slots. Default value is **Off**.
RX User Clock Setting		
Enable RX user clock	**On/Off**	Divider values of RX CDR output frequency. If the clock is not used, you can disable the clock to save power. This clock source drives both RX **User Clock1** and **User Clock 2** in the Core Interface. Default value is **Off**.
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**.

Figure 67. RX FHT PMA Parameters in Parameter Editor

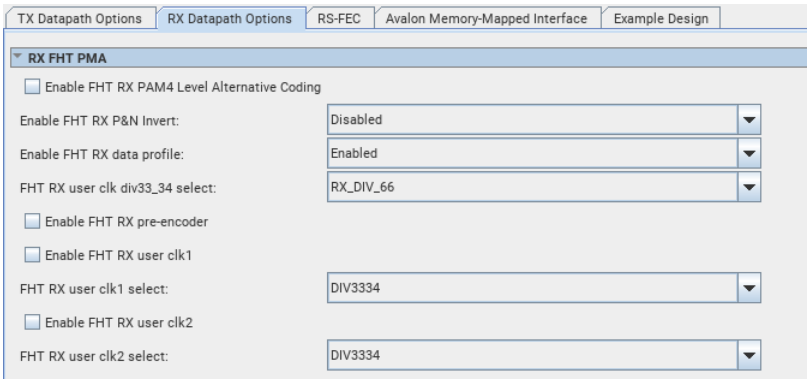


Table 33. RX FHT PMA Parameters

Parameter	Values	Description
Enable FHT RX PAM4 Level Alternative Coding	**On/Off**	Enable this for RX PAM4 Level Alternative Coding. When disabled, link partner must send gray code set to 0xB4. When enabled, link partner must send gray code set to 0x6C. You must disable this parameter for normal operation or when in internal or external loopback. Default value is **Off**.
Enable FHT RX P&N Invert	**Disabled/Enabled**	Enable this to invert RX P and N input. Default is **Disabled**.
Enable FHT RX data profile	**Disabled/Enabled**	Enable FHT RX data profile to set the threshold for number of 1's in 1M RX Data bits that determine the quality of RX data. If the number of 1's received is not within the specified min and max threshold, then RX bad status is indicated. You determine the threshold minimum and maximum based on your data and specify it in **FHT RX data 1 count maximum** and **FHT RX data 1 count minimum** parameters described below. Default is **Disabled**. *Note:* This parameter must be **Enabled**.
FHT Rx data 1 count maximum	550000	Selects high threshold of 1s in 1M RX data bits. Default is 550000. Recommended value is 550000. Must be used in conjunction with the **Enable FHT RX data profile** parameter.
FHT Rx data 1 count minimum	450000	Selects low threshold of 1s in 1M RX data bits. Default is 450000. Recommended value is 450000. Must be used in conjunction with the **Enable FHT RX data profile** parameter.
FHT RX user clk div33_34 select	**RX_DIV_33** **RX_DIV_34** **RX_DIV_66** **RX_DIV_68**	Selects one of the four DIV clock output for the RX user clock. Refer to Clocking on page 134. Default is **RX_DIV_66**.

continued...

Parameter	Values	Description
Enable FHT RX pre-encoder	**On/Off**	Enables FHT TX pre-encoder. Default value is off. This setting must match the link partner's RX pre-encoder setting.
Enable FHT RX user clk1	**On/Off**	Enables FHT RX user clk1. Default is **Off**.
FHT RX user clk1 select	**DIV3334** **DIV40**	FHT RX user clk1 select. Off selects **div3334** (one of the four DIV clocks listed in user div33_34). On selects **DIV40** clock. Refer to Clocking on page 134. Default is **div3334**.
Enable FHT RX user clk2	**On/Off**	Enables FHT RX user clk2. Default value is **Off**.
FHT RX user clk2 select	**DIV3334** **DIV40**	FHT RX user clk2 select. Off selects **div3334** (one of the four DIV clocks listed in user div33_34). On selects **DIV40** clock. Refer to Clocking on page 134. Default is **div3334**.

3.3.3.1. RX FGT PMA Interface Options

Figure 68. RX FGT PMA Interface Options in Parameter Editor

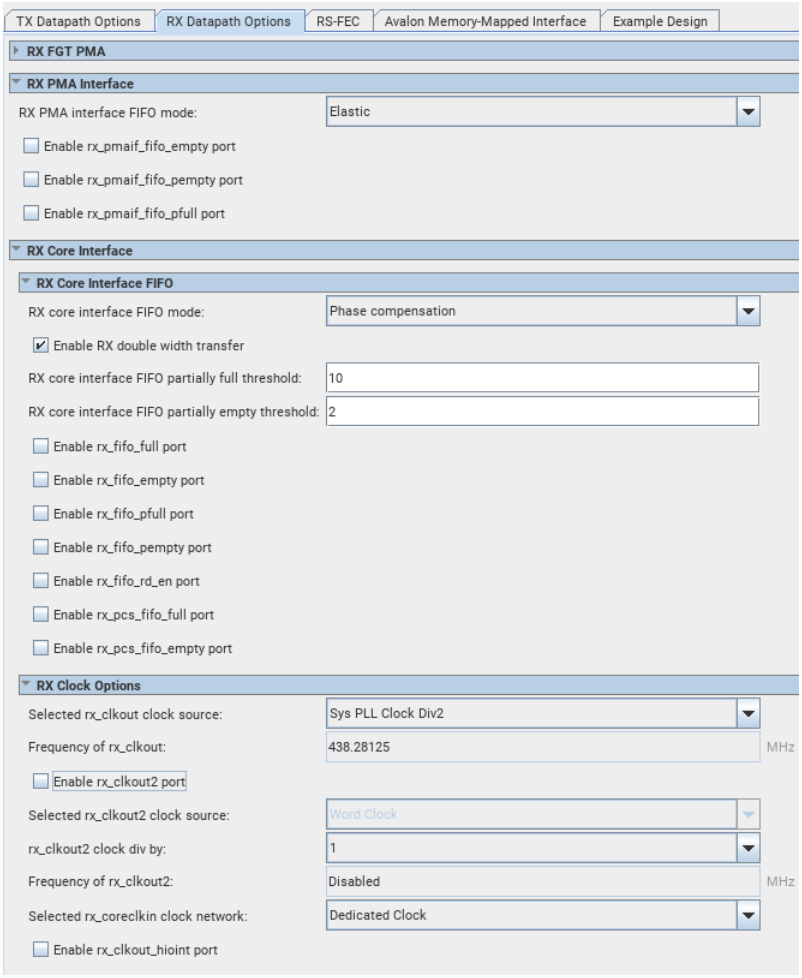


Table 34. RX FGT PMA Interface Parameters

Parameter	Values	Description
RX FGT PMA Parameters		
RX PMA interface FIFO mode	**Register** **Elastic**	Selects the RX PMA Interface FIFO mode. Default value is **Elastic**.
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**.

continued...

Parameter	Values	Description
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**.
RX core interface FIFO partially full threshold	10	Specifies the partially full threshold for the RX Core Interface FIFO. Default value is 10.
RX core interface FIFO partially empty threshold	2	Specifies the partially empty threshold for the RX Core Interface FIFO. Default value is 2.
Enable rx_fifo_full port	**On/Off**	Enables the optional rx_fifo_full status output port. This signal indicates when the RX core FIFO has reached the full threshold. This signal is synchronous with `rx_clkout`. Default value is **Off**.
Enable rx_fifo_empty port	**On/Off**	Enables the optional `rx_fifo_empty` status output port. This signal indicates when the RX core FIFO has reached the empty threshold. This signal is synchronous with `rx_clkout`. Default value is **Off**.
Enable rx_fifo_pfull port	**On/Off**	Enables the optional `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 `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 `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** **Bond Clock** **User Clock 1** **User Clock 2** **Sys PLL Clock** **Sys PLL Clock Div2**	Specifies the `rx_clkout` output port source. Default value is **Sys PLL Clock Div2**.
Frequency of rx_clkout	Output	Displays the frequency of `rx_clkout` in MHz based on `rx_clkout` source selection.
Enable rx_clkout2 port	**On/Off**	Enables the optional `rx_clkout2` output clock. Default value is **Off**.
Selected rx_clkout2 clock source	**Word Clock** **Bond Clock** **User Clock 1** **User Clock 2** **Sys PLL Clock** **Sys PLL Clock Div2**	Specifies the `rx_clkout` output port source. Default value is **Word Clock**.

continued...

Parameter	Values	Description
rx_clkout2 clock div by	**1**, **2**	Selects the rx_clkout2 divider setting that divides out the `rx_clkout2` output port source. Default value is **1**.
Frequency of rx_clkout2	Output	Displays the frequency of `rx_clkout2` in MHz based on `rx_clkout2` source selection and `rx_clkout2` clock divide by factor.
Selected rx_coreclkin clock network	**Dedicated Clock** **Global Clock**	Specifies the type of clock network to use to route the clock signal to `rx_coreclkin` port. **Dedicated Clock** allows a higher maximum frequency (fmax) between the FPGA fabric and the FPGA fabric and RX Core interface FIFO. The number of **Dedicated Clock** lines are limited. Default value is **Dedicated Clock**.

3.3.4. RS-FEC (Reed Solomon Forward Error Correction) Options

The F-Tile PMA/FEC Direct PHY Intel FPGA IP supports RS-FEC (528, 514), RS (544, 514), RS (272, 258). You can enable this functionality in the parameter editor by selecting the **Enable RS-FEC** option on the **RS-FEC** tab under **Common Datapath Options**.

F-Tile PMA/FEC Direct PHY Intel FPGA IP is available in 25G FEC as a building block, which means the smallest module for FEC is one 25G. You must ensure that clock and reset signals are shared from the same 100G FEC core where they implement the IP.

When you turn on the **Enable RS-FEC** option for the F-Tile PMA/FEC Direct PHY Intel FPGA IP uses the RS-FEC block, even if it uses only one channel in the IP. You can use the same F-Tile PMA/FEC Direct PHY Intel FPGA IP core to implement different protocols. You can enable RS-FEC and TX/RX options independently. However, the FEC mode must be the same. If **Enable RS-FEC** if off, all the options below are grayed out.

There are 32-bit CWBIN counters that are implemented in soft IP. The soft logic converts the 8-bit CWBIN 0-3 register in the FEC block of the Hard IP to 32-bit soft logic registers. You can enable the 32-bit CWBIN counters using the parameter settings and they are available for all FEC modes.

Note: For more information on how to access the 32-bit CWBIN registers, refer to PMA and FEC Direct PHY Soft CSR Register Map. For more information on how to to access the other FEC registers, please refer to FEC Register Map.

The F-Tile PMA/FEC Direct PHY Intel FPGA IP supports the following modes:

- Ethernet Technology Consortium* (ETC) RS (272,258)
- IEEE 802.3 RS (528,514) (CL 91)
- IEEE 802.3 RS (528,514) (CL 91) ETC
- Fibre Channel RS (528, 514)
- FlexO RS (528, 514)
- IEEE 802.3 RS (544,514) (CL 134)
- Custom IEEE 802.3 RS (544, 514) (CL 134) @26.5625Gbps

- Interlaken RS (544, 514)
- Fibre Channel RS (544, 514)
- FlexO RS (544, 514)

Figure 69. RS-FEC Options in Parameter Editor

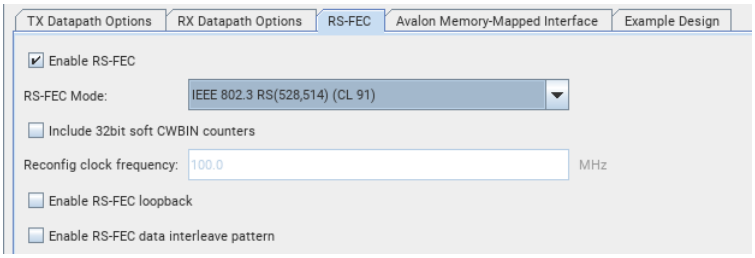


Table 35. RS-FEC Parameters

Parameter	Values	Description
Enable RS-FEC	On/Off	Enables the RS-FEC module. Default value is **Off**. *Note:* When the **Enable RS-FEC** option is on, a separate interface is not available for each PMA by use of the **Provide separate interface for each PMA** option.
RS-FEC Mode	• **Ethernet Technology Consortium (ETC) RS (272,258)** • **IEEE 802.3 RS (528,514) (CL 91)** • **IEEE 802.3 RS (528,514) (CL 91) ETC** • **Fibre Channel RS (528, 514)** • **FlexO RS (528, 514)** • **IEEE 802.3 RS (544,514) (CL 134)** • **Custom IEEE 802.3 RS (544, 514) (CL 134) @26.5625Gbps** • **Interlaken RS (544, 514)** • **Fibre Channel RS (544, 514)** • **FlexO RS (544, 514)**	Specifies the RS-FEC mode for various topologies. Default value is **IEEE 802.3 RS (528,514) (CL 91)**.
Include 32bit soft CWBIN counters	On/Off	Enables soft implementation of the 32-bit CWBIN 0-3 counters. This parameter is available only when RS-FEC is enabled and greyed out when RS-FEC is disabled.
Reconfig clock frequency	100MHz to 250MHz	Available only when 32-bit soft CWBIN counters are enabled. The reconfig clock frequency that you are using should be provided here.
Enable RS-FEC loopback	On/Off	Enables loopback for RS-FEC.
Enable RS-FEC Data interleave pattern	On/Off	FEC lanes are bit-interleaved on each physical lane. When enabled: 64/80 (only for IEEE 802). Default value is **Off**.

3.3.5. Avalon Memory Mapped Interface Options

Figure 70. Avalon Memory Mapped Interface Options Tab in Parameter Editor

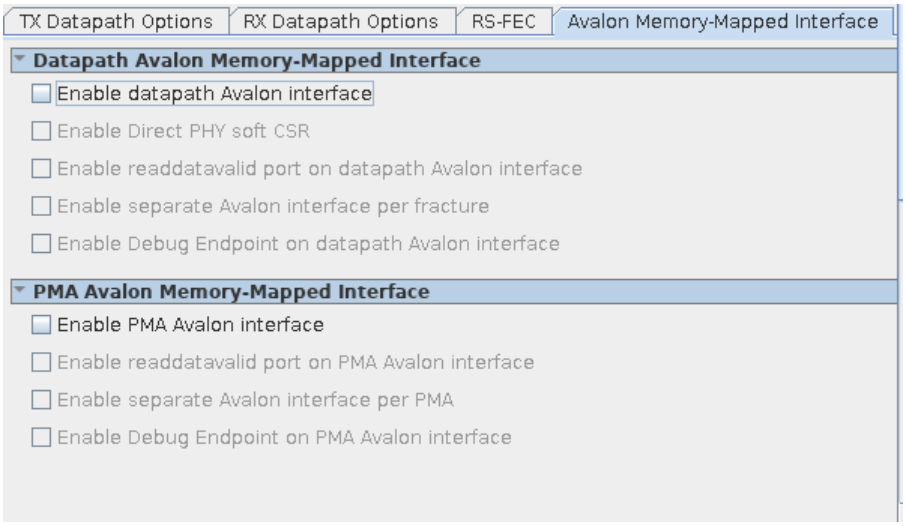


Table 36. Avalon Memory Mapped Interface Parameters

Parameter	Values	Description
Enable datapath Avalon Interface	**On/Off**	Enables or disables the datapath Avalon 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 datapath Avalon interface	**On/Off**	**Off** specifies no `readdatavalid` port, `waitrequest` low indicates data valid. **On** specifies `readdatavalid` port indicates data valid. Default value is **Off**.
Enable separate datapath Avalon interface per fracture	**On/Off**	**Off** specifies shared interface. **On** specifies split interface, if multiple interfaces available with selected targets. Default value is **Off**.
Enable Debug Endpoint on datapath Avalon interface	**On/Off**	When **On**, the F-Tile 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**.
Enable PMA Avalon interface	**On/Off**	Enables or disables the PMA Avalon interface. Default value is **Off**.
Enable readdatavalid port on PMA Avalon interface	**On/Off**	**Off** specifies no `readdatavalid` port, `waitrequest` low indicates data valid. **On** specifies port `readdatavalid` indicates data valid. Default value is **Off**.
Enable separate PMA Avalon interface per PMA	**On/Off**	**Off** specifies shared interface.

continued...

Parameter	Values	Description
		On specifies split interface if there are multiple PMA in a system.
Enable Debug Endpoint on PMA Avalon interface	**On/Off**	When enabled, the direct PHY IP includes an embedded Debug Endpoint that internally connects to the Avalon memory-mapped agent interface. The Debug Endpoint can access the reconfiguration space of the PMA. It can perform certain test and debug functions via JTAG using the System Console. This option may require that you include a `jtag_debug` link in the system. Default value is **Off**.

3.3.6. Register Map IP-XACT Support

When you generate the F-Tile PMA/FEC Direct PHY Intel FPGA IP using Intel Quartus Prime Pro Edition software version 22.2 or later, the IP-XACT information for the IP is included in the `<ip_name>.ip` file. The generated IP-XACT information includes the register map about your IP. If your design uses FGT PMA, then the FGT register map is included in the `<ip_name>.ip` file and similarly for FHT PMA and soft CSR registers.

Use the following steps to generate the register map information in IP-XACT format:

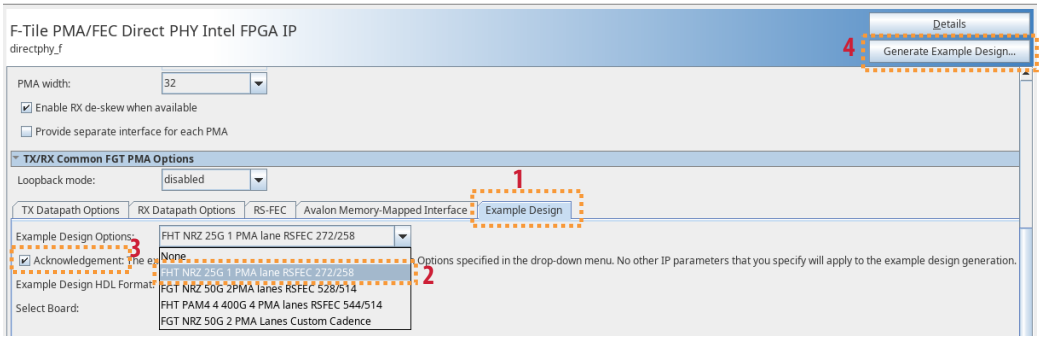
1. In the F-Tile PMA/FEC Direct PHY Intel FPGA IP, enable the **Enable datapath Avalon interface**, **Enable Direct PHY soft CSR** and **Enable PMA Avalon Interface** in the **Avalon Memory-Mapped Interface** tab.
2. Click **Generate** and check your `<ip_name>.ip` file.

3.3.7. Example Design Generation

The F-Tile PMA/FEC Direct PHY Intel FPGA IP parameter editor includes the **Generate Example Design** function to easily create, generate, and simulate the PMA/FEC direct mode example design.

You can select from four **Example Design Options** for generation, as Example Design Options in IP Parameter Editor shows.

Figure 71. Example Design Options in IP Parameter Editor



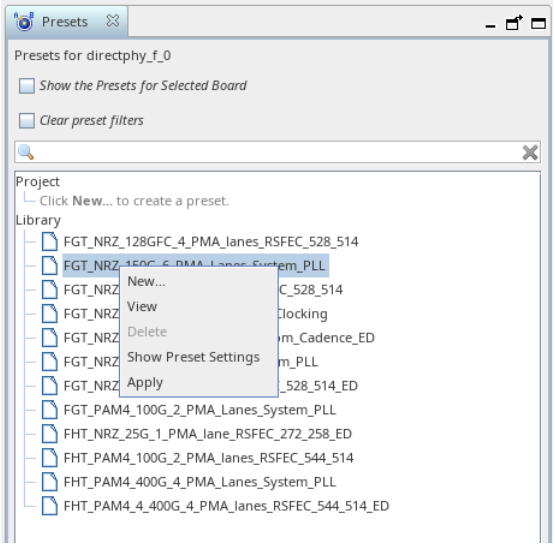
The example designs support generation, compilation, and simulation flows for the target device. Starting in Intel Quartus Prime software version 22.1, hardware support for example designs is enabled in the Intel Agilex 7 I-Series Transceiver-SoC development kit. The following **Example Design Options** are currently available:

Table 37. Example Design Generation Options

Example Design Options	Preset Setting Equivalent	Description
FHT NRZ 25G 1 PMA lane RSFEC 272/258	**FHT_NRZ_25G_1_PMA_Lane_RSFEC_272_258_ED**	1 PMA FHT NRZ lane operating at 25.78125 Gbps with RS-FEC 272/258 mode.
FGT NRZ 50G 2 PMA lanes RSFEC 528/514	**FGT_NRZ_50G_2_PMA_Lanes_RSFEC_528_514_ED**	2 PMA FGT NRZ lanes operating at 25.78125 Gbps (each lane) with RS-FEC 528/514 mode.
FHT PAM4 4 400G 4 PMA lanes RSFEC 544/514	**FHT_PAM4_400G_4_PMA_lanes_RSFEC_544_514_ED**	4 PMA FHT PAM4 lanes operating at 106.25 Gbps (each lane) with RS-FEC 544/514 mode.
FGT NRZ 50G 2 PMA Lanes Custom Cadence	**FGT_NRZ_50G_2_PMA_Lanes_Custom_Cadence_ED**	2 PMA FGT NRZ lanes operating at 25.78125 Gbps (each lane) with custom cadence clocking mode. In custom cadence clocking mode, the system PLL clocks the digital data path (that is, the F-tile interface FIFO and core interface FIFO) of the PMA. The PMA block and PMA interface FIFO is clocked by PMA clockout.

The **Example Design Options** are equivalent with some of the preset settings, as Example Design Generation Options describes. To review the IP parameter settings for each preset, refer to F-Tile PMA/FEC Direct PHY Intel FPGA IP Available Parameter Presets. Alternatively, right-click a preset in the IP parameter editor, and then click **Show Preset Settings**, or click **Apply preset** to apply the preset's settings in the parameter editor.

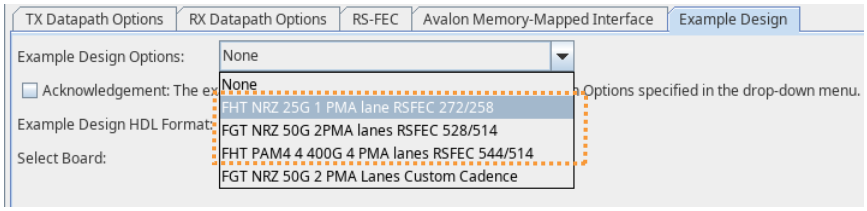
Figure 72. Show Preset Settings



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

The **Example Design** tab of the F-Tile PMA/FEC Direct PHY Intel FPGA IP allows you to select pre-defined RS-FEC options to configure an example design as shown in the following figure.

Figure 73. RS-FEC Example Designs in the F-Tile PMA/FEC Direct PHY Intel FPGA IP



There are three RS-FEC example designs available in the F-Tile PMA/FEC Direct PHY Intel FPGA IP as shown:

- **FHT NRZ 25G 1 PMA Lane RSFEC 272/25**
- **FGT NRZ 50G 2 PMA Lanes RSFEC 528/514**
- **FHT PAM4 4 400G 4 PMA Lanes RSFEC 544/514**

All the example designs follow the configuration options for the FEC direct mode as described in the following sections:

- TX Datapath Options
- RX Datapath Options
- Avalon Memory Mapped Interface Options

To generate an example design, follow the below steps:

1. Go to the **Example Design** tab in the F-Tile 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 options is to remind you that only the example design you specify in the drop-down menu is generated. No other IP parameters setting that you specify take effect in the example design generation. 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 **Generate Example Design**.

Clicking **Generate Example Design** completes the **IP Generation** and **Support-logic Generation** stages of the Compiler. An example design folder also generates containing the Intel Quartus Prime project (`.qpf`), settings (`.qsf`), and IP files, simulation, and testbench files for the example design in the following location:

```
<Project Folder>/<directphy_f_0_example_design/example_design>
```

The Compiler reads the example design `.qsf` file that contains the PMA reference clock, and the TX and RX high speed serial pin location assignments.

In order to provide a reduction in real-time simulation duration, the example design testbench uses a *Fast Sim* model. This model is enabled via a macro in the simulation run scripts. The syntax to enable the *Fast Sim* model is as follows:

```
+define+IP7581SERDES_UX_SIMSPEED
```

This macro is enabled by default in the example design simulation scripts after you click **Generate Example Design** button.

You can use the *Fast Sim* model when you are simulating the F-Tile PMA/FEC Direct PHY Intel FPGA IP design to reduce simulation time. However, in order to use the *Fast Sim* model, ensure all the IPs in your design that you place within the same F-tile, support *Fast Sim* mode. For example; if you have a PMA direct mode design together with other protocol IPs that do not support the *Fast Sim* mode and place them within the same F-tile, you can run into simulation errors for your PMA direct design.

Note: This macro is not available when you use FGT cascade mode, FGT dual simplex mode or when you use the FHT PMA.

Example Design Simulation

To simulate the example design in the VCS, VCS MX, ModelSim, or Xcelium simulators, use the following commands. The RTL files are in the `example_design/rtl` directory and simulation files are in the `example_design/testbench` directory.

- To simulate with VCS, go to the `example_design/testbench` directory and the launch the simulation using the shell script:

  ```
  sh run_vcs.sh
  ```

- To simulate with VCS MX, go to the `example_design/testbench` directory and the launch the simulation using the shell script:

  ```
  sh run_vcsmx.sh
  ```

- To simulate with ModelSim, go to the `example_design/testbench` directory and the launch the simulation using the command:

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

- To simulate with Xcelium, go to the `example_design/testbench` directory and the launch the simulation using the shell script:

  ```
  sh run_xcelium.sh
  ```

- Launch waveform viewer to see the simulation results.

Starting in Intel Quartus Prime software version 21.3, the example designs support VHDL with VCS MX and ModelSim simulators.

3.4. Signal and Port Reference

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

Each `tx_parallel_data` and `rx_parallel_data` bus is exposed as 80 to 320 bits. Some bits map to special functionality.

Each PMA channel transmits and receives 80 to 320 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 on page 124.

When you enable the **Provide separate interface for each PMA** option for the F-Tile 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:

`tx_serial_data_xcvr0`, `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: `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:

- `system_pll_clk_link`, `rx_cdr_divclk_link0`, `rx_cdr_divclk_link1`
- `tx_reset`, `rx_reset`, `tx_reset_ack`, `rx_reset_ack`, `tx_ready`, `rx_ready`
- `rsfec` signals
- `tx_cadence`, `tx_cadence_fast_clk`, `tx_cadence_slow_clk`, `tx_cadence_slow_clk_locked`

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

When **Number of system copies** index is more than 1 (from 2-16), then the PMA lane signal names are appended with a `_sys<n>` suffix, with *n = PMA* index number. The following are the only signals that are not appended with the `_sys` suffix.

- Reconfiguration Avalon memory-mapped interface ports
- `rx_cdr_divclk_link0`, and `rx_cdr_divclk_link1`

Table 38. Variables Defining Bits for the Interfacing Ports in Port and Signal Reference

Variable	Values	Description
<*N*>	FGT: 1, 2, 4, 6, 8, 12, 16 FHT: 1, 2, 4	*N* is the number of PMA lanes.
<*n*>	`0 to N-1`	*n* is the PMA index number.
<*X*>	PMA width = 8, 10, 16, 20, and 32-bit, *X*=1 PMA width = 64-bit, *X*=2	*X* is the number of streams.

continued...

Variable	Values	Description
	PMA width = 128-bit, *X*=4	
<K_p>	`Ceiling(log2(N))` K_p = 0,1,2,3,3,4,4 for *N* = 1,2,4,6,8,12,16	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.
<K_d>	`Ceiling(log2(N))` K_d = 0,1,2,3,3,4,4 for *N* = 1,2,4,6,8,12,16	K_d is the datapath reconfiguration interface address. K_d=0 if a separate Avalon interface per PMA is enabled or FEC is enabled. K_d=`Ceiling(log2(N)` if separate Avalon interface per PMA is disabled and FEC is disabled.
<*D*>	If PMA width = 8, 10, 16, 20, or 32-bit, then *D* = **PMA Width** If PMA width = 64 or 128-bit, then *D* = **32**	*D* is the data width value to calculate the total parallel data bits.

3.4.1. TX and RX Parallel and Serial Interface Signals

Table 39. 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
`tx_parallel_data [(80 * N * X)-1:0]`	`tx_coreclkin` `tx_reset`	input	Parallel data bus from FPGA core to F-tile interface. Some bits map to specific functionality, as Parallel Data Mapping Information on page 124 describes.
`rx_parallel_data[(80 * N * X) -1:0]`	`rx_coreclkin` `rx_reset`	output	Parallel data bus from FPGA core to F-tile interface. Some bits map to specific functionality, as TX and RX Parallel Data Mapping Information for Different Configurations on page 126 describes.
`tx_serial_data [N-1:0]`	`tx_reset`	output	TX serial data port.
`tx_serial_data_n [N-1:0]`	`tx_reset`	output	Differential pair for TX serial data port.
`rx_serial_data [N-1:0]`	`rx_reset`	input	RX serial data port.
`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 40. TX and RX Reference Clock and Clock Output 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

`rx_clkout [(N*X)-1:0]`
`rx_clkout2 [(N*X)-1:0]]`
`tx_clkout [(N*X)-1:0]`
`tx_clkout2 [(N*X)-1:0]`
	N/A	output	Refer to Clock Ports on page 138
			Note: It is recommended to always use bit[0] to drive `tx_coreclkin[N*X-1:0]` and `rx_coreclkin[N*X-1:0]`. When *X* is larger than 1, bit [((*n*+1)*X)-1: (*n**X)+1] does not have valid output and must not be used.
			For example, when PMA width = 64, *X* = 2:
			• If *N*=1, *n*=0: bit 1 does not have valid output and must not be used.
			• If *N*=8, *n*=0 to 7: bits 1, 3, 5, 7, 9, 11, 13, 15 do not have valid output and must not be used.
			when PMA width = 128, *X* = 4:
			• If *N*=1, *n*=0: bits 1, 2, 3 do not have valid output and must not be used.
			• If *N*=4, *n*=0 to 3: bits 1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15 do not have valid output and must not be used.

`tx_coreclkin [N*X-1:0]`	N/A	input	The FPGA core clock. Drives the write side of the TX FIFO.
`rx_coreclkin [N*X-1:0]`	N/A	input	The FPGA core clock. Drives the read side of the RX FIFO.
`tx_pll_refclk_link [N-1:0]`(26) *Note:* This signal is single bit when **Enable TX FGT PLL cascade mode** is enabled.	N/A	input	This is neither physical nor logical pin. You connect this to <`out_refclk_fgt_X`> port from the F-Tile Reference and System PLL Clocks Intel FPGA IP(27).
`rx_cdr_refclk_link [N-1:0]` *Note:* This signal is not available when **Enable TX FGT PLL cascade mode** is enabled.	N/A	input	This is neither physical nor logical pin. You connect this to <`out_refclk_fgt_X`> port from the F-Tile Reference and System PLL Clocks Intel FPGA IP(27).
`system_pll_clk_link`	N/A	input	This is neither physical nor logical pin. You connect this to <`out_systempll_clk_0`> port from the F-Tile Reference and System PLL Clocks Intel FPGA IP(27).

continued...

(26) Ports ending in "_link" must connect to the F-Tile Reference and System PLL Clocks Intel FPGA IP. These ports cannot be simulated.

(27) Refer to Guidelines for F-Tile Reference and System PLL Clocks Intel FPGA IP Usage on page 195 for reference clock and system PLL usage.

Signal Name	Clocks Domain/ Resets	Direction	Description
tx_pll_locked [*N*-1:0]	asynchronous	output	TX PLL locked signal for both FGT and FHT to reference clock within the PPM threshold status signal. 1'b1 = locked. 1'b0 = not locked.
rx_cdr_divclk_link0	N/A	output	Clock output from FGT CDR divided clock. This signal is used for CPRI. F-tile includes a total of two such pins. This port is neither physical nor logical pin. If you enable, you must set the number of system copies to 1. This port must connect to the out_cdrclk port of the F-Tile Reference and System PLL Clocks Intel FPGA IP . This port cannot be enabled in a quad that has primary PLL configuration(27).
rx_cdr_divclk_link1	N/A	output	Clock output from FGT CDR divided clock. This signal is used for CPRI. In whole F tile, there are 2 of such pins. This port is neither physical nor logical pin. If you enable this port, you must set number of system copies to 1. This port must connect to the out_cdrclk port of F-Tile Reference and System PLL Clocks Intel FPGA IP. You cannot enable a quad that has primary PLL configuration in PLL cascade mode. Not supported for FHT(27).

3.4.3. Reset Signals

Table 41. Reset Signals

Signal Name	Clocks Domains	Direction	Description
tx_reset	asynchronous	input	TX reset input for TX PMA and TX datapath. Must asserted until tx_reset_ack is asserted.
rx_reset	asynchronous	input	RX reset input for RX PMA and RX datapath. Must be kept asserted until rx_reset_ack is asserted.
tx_reset_ack	asynchronous	output	TX fully in reset indicator. This signal asserts following tx_reset assertion and stays asserted for as long as tx_reset is asserted. This signal deasserts following tx_reset deassertion and remains deasserted for as long as tx_reset is deasserted.
rx_reset_ack	asynchronous	output	RX fully in reset indicator. This signal asserts following rx_reset assertion and stays asserted for as long as rx_reset is asserted. This signal deasserts following rx_reset deassertion and remains deasserted for as long as rx_reset is deasserted.
tx_am_gen_start	asynchronous	output	When using FEC, indicates when to start sending alignment markers. This clears once tx_am_gen_2x_ack is asserted.

continued...

Signal Name	Clocks Domains	Direction	Description
tx_am_gen_2x_ack	asynchronous	input	When using FEC, indicates to the reset sequencer at least 2 alignment markers were sent since tx_am_gen_start is asserted. This signal is deasserted after tx_am_gen_start is deasserted.
tx_ready	asynchronous	output	Status port to indicate when TX PMA and TX datapath are reset successfully and ready for data transfer.
rx_ready	asynchronous	output	If RX de-skew is disabled: Status port to indicate when RX PMA and RX datapath are reset successfully and ready for data transfer. If RX de-skew is enabled: Status port to indicate when RX PMA and RX datapath are reset successfully, RX de-skew is done, and ready for data transfer. *Note:* During F-Tile link initialization, the data pattern sent from the TX has to be scrambled to get rx_ready to assert. If a 0101 pattern or other constant patterns are sent, rx_ready does not assert, and the link does not initialize.

3.4.4. RS-FEC Signals

Table 42. RS-FEC Signals

Signal Name	Clocks Domain/ Resets	Direction	Description
rsfec_status_rx_not_deskew	asynchronous	output	All RX lanes locked but the alignment markers were not unique or the skew was too large. Only applicable in multi-lane.
rsfec_status_rx_not_locked	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.
rsfec_status_rx_not_align	asynchronous	output	Incoming signal fail, RX lanes not all locked, alignment markers not unique or skew too large. Only applicable in multi-lane.
rsfec_sf	asynchronous	output	Signal fail, low means RS-FEC is aligned(fec_ready is high and rsfec_status_not_aligned is low)
fec_snapshot	asynchronous	input	Takes a snap of RS-FEC status to CSR, uses Avalon memory-mapped to read the content. To avoid a SSR variation delay between different streams in aggregate mode for RS-FEC error counters across multiple streams, stop traffic before taking the snapshot.

3.4.5. Custom Cadence Control and Status Signals

Table 43. Custom Cadence Control and Status Signals

Signal Name	Clocks Domain/ Resets	Direction	Description
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/bond 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 on page 124.
tx_cadence_fast_clk	N/A	input	Fast clock input for tx_cadence generator. Use this as the system clock within F-tile (or use (system clock)/2 when Core Interface is in double width mode). Refer to Custom Cadence Generation Ports and Logic on page 145.
tx_cadence_slow_clk	N/A	input	Slow clock input for tx_cadence generator. Use this clock as the PMA word/bond clock (or (PMA word/bond clock)/2 when Core Interface is in double width mode). Refer to Custom Cadence Generation Ports and Logic on page 145.
tx_cadence_slow_clk_locked	N/A	input	By default, CCG logic assumes tx_cadence_slow_clk_locked is coming from TX PLL, and uses tx_pll_locked to deassert CGG logic reset. However, if tx_cadence_slow_clk is not directly coming from the TX PLL word clock/bond 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. 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. TX PMA Control Signals

Table 44. TX PMA Control Signals

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

Signal Name	Clock Domain/ Resets	Direction	Description
fgt_tx_beacon[*N*-1:0]	asynchronous	input	1'b1: enable SATA beacon signal. 1'b0: disable SATA beacon signal.
fgt_tx_pma_elecidle[(3**x*-1:0)]	tx_coreclkin tx_reset	input	When the **FGT PMA configuration rules** parameter is set to SATA or USB and **Enable simplified TX data interface** parameter is enabled, this port is available as a separate 4-bit bus. When asserted, the FGT PMA transmitter is forced into an electrical idle condition. • 4'b0000: FGT TX is not in electrical idle mode. • 4'b1111: FGT TX enters electrical idle mode.

3.4.7. RX PMA Status Signals

Table 45. RX PMA Status 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
`fgt_rx_signal_detect [N-1:0]`	asynchronous	output	FGT RX signal detect indicator.
`fgt_rx_signal_detect_lfps [N-1:0]`	asynchronous	output	Indicates SATA or USB low frequency periodic signaling (LFPS) signal detection.
`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 FGT PMA only When `lockedtodata` stays high, the `lockedtoref` signal status is insignificant.
`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 both FGT and FHT PMA. 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.
`fgt_rx_set_locktoref [N-1:0]`	asynchronous	input	`1'b1`: keep CDR in lock to reference mode. `1'b0`: keep CDR in auto mode.
`fgt_rx_cdr_freeze[N-1: 0]`	asynchronous	input	This port is used in GPON to freeze the CDR lock state during inactive time-slots. • `1'b1`: Freeze the CDR • `1'b0`: Unfreeze the CDR
`fgt_rx_cdr_fast_freeze_sel[N-1: 0]`	asynchronous	input	This port is used in GPON for CDR freeze signal select. For GPON mode, you must tie this signal to `1'b0`.

3.4.8. TX and RX PMA and Core Interface FIFO Signals

Table 46. TX and RX PMA and Core Interface FIFO 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
`tx_pmaif_fifo_empty [(N*X)-1:0]`	asynchronous	output	PMA Interface TX FIFO empty.
`tx_pmaif_fifo_pempty [(N*X)-1:0]`	asynchronous	output	PMA Interface TX FIFO partially empty.
`tx_pmaif_fifo_pfull [(N*X)-1:0]`	asynchronous	output	PMA Interface TX FIFO partially full.
`rx_pmaif_fifo_empty [(N*X)-1:0]`	asynchronous	output	PMA Interface RX FIFO empty.
`rx_pmaif_fifo_pempty [(N*X)-1:0]`	asynchronous	output	PMA Interface RX FIFO partially empty.
`rx_pmaif_fifo_pfull [(N*X)-1:0]`	asynchronous	output	PMA Interface RX FIFO partially full.

continued...

Signal Name	Clocks Domain/Resets	Direction	Description
tx_fifo_full [(N*X)-1:0]	tx_coreclkin tx_reset	output	Core Interface TX FIFO full port.
tx_fifo_empty [(N*X)-1:0]	TX Word Clock TX Bond Clock Sys PLL Clock	output	Core Interface TX FIFO empty port.
tx_fifo_pfull [(N*X)-1:0]	tx_coreclkin tx_reset	output	Core Interface TX FIFO partially full port.
tx_fifo_pempty [(N*X)-1:0]	TX Word Clock TX Bond Clock Sys PLL Clock	output	Core Interface TX FIFO partially empty port.
rx_fifo_full [(N*X)-1:0]	Transfer clock: Word Clock Bond Clock Sys PLL Clock rx_reset	output	Core Interface RX FIFO full port.
rx_fifo_empty [(N*X)-1:0]	rx_coreclkin rx_reset	output	Core Interface RX FIFO empty port.
rx_fifo_pfull [(N*X)-1:0]	Transfer clock: Word Clock Bond Clock Sys PLL Clock rx_reset	output	Core Interface RX FIFO partially full port.
rx_fifo_pempty [(N*X)-1:0]	rx_coreclkin rx_reset	output	Core Interface RX FIFO partially empty port.
tx_dll_lock [(N*X)-1:0]	tx_reset	output	TX DLL locked status signal for data transfer. Monitor this signal when the core interface FIFO is in elastic mode, wait for tx_dll_lock port to assert before asserting the write enable bit for the core interface FIFO. Refer to TX and RX Parallel Data Mapping Information for Different Configurations on page 126.
rx_fifo_rd_en [(N*X)-1:0]	rx_coreclkin rx_reset	input	Core Interface RX FIFO read enable port.

3.4.9. PMA Avalon Memory Mapped Interface Signals

Table 47. PMA Avalon Memory Mapped Interface Signals (Enable Separate Avalon Interface per PMA = 0)

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

Signal Name	Clocks Domain/Resets	Direction	Description
reconfig_xcvr_clk	Clock	Input	Reconfig Interface Clock. Intel recommends a frequency of 100 to 250 MHz for this clock.
reconfig_xcvr_reset	reconfig_xcvr_clk	Input	Reconfig Interface Reset

continued...

Signal Name	Clocks Domain/Resets	Direction	Description
reconfig_xcvr_address[17+K_p:0]	reconfig_xcvr_clk	Input	Reconfig Interface Address K_p=Ceiling(log2(*N*)). Upper address bits are for shared PMA decoding if more than one PMA exists.
reconfig_xcvr_byteenable [3:0]	reconfig_xcvr_clk	Input	Byte Enable. If byteenable[3:0] is 4'b1111, uses 32-bit Dword Access; otherwise uses byte access.
reconfig_xcvr_write	reconfig_xcvr_clk	Input	Reconfig Write
reconfig_xcvr_read	reconfig_xcvr_clk	Input	Reconfig Read
reconfig_xcvr_writedata [31:0]	reconfig_xcvr_clk	Input	Reconfig Write data
reconfig_xcvr_readdata [31:0]	reconfig_xcvr_clk	Output	Reconfig Read data
reconfig_xcvr_waitrequest	reconfig_xcvr_clk	Output	Reconfig Wait Request
reconfig_xcvr_readdatavalid	reconfig_xcvr_clk	Output	Reconfig Read Data Valid. Optional port, available if the port is enabled in parameter editor.

Table 48. PMA Avalon Memory Mapped Interface Signals (Enable Separate Avalon Interface per PMA = 1)

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

Signal Name	Clocks Domain/Resets	Direction	Description
reconfig_xcvr<*n*>_clk	Clock	Input	Reconfig Interface Clock. Intel recommends a frequency of 100 to 250 MHz for this clock.
reconfig_xcvr<*n*>_reset	reconfig_xcvr<*n*>_clk	Input	Reconfig Interface Reset
reconfig_xcvr<n>_address[17:0]	reconfig_xcvr<*n*>_clk	Input	Upper address bits are for shared PMA decoding if more than 1 PMA exists.
reconfig_xcvr<*n*>_byteenable [3:0]	reconfig_xcvr<*n*>_clk	Input	Byte Enable. If byteenable[3:0] is 4'b1111, uses 32-bit Dword; otherwise uses byte access.
reconfig_xcvr<*n*>_write	reconfig_xcvr<*n*>_clk	Input	Reconfig Write
reconfig_xcvr<*n*>_read	reconfig_xcvr<*n*>_clk	Input	Reconfig Read
reconfig_xcvr<n>_writedata [31:0]	reconfig_xcvr<*n*>_clk	Input	Reconfig Write data
reconfig_xcvr<n>_readdata [31:0]	reconfig_xcvr<*n*>_clk	Output	Reconfig Read data
reconfig_xcvr<*n*>_waitrequest	reconfig_xcvr<*n*>_clk	Output	Reconfig Wait Request
reconfig_xcvr<*n*>_readdatavalid	reconfig_xcvr<*n*>_clk	Output	Reconfig Read Data Valid. Optional port, available if the port is enabled in parameter editor.

3.4.10. Datapath Avalon Memory Mapped Interface Signals

Table 49. Datapath Avalon Memory Mapped Interface Signals (Enable Separate Avalon Interface per Fracture = 0)

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

Signal Name	Clocks Domain/Resets	Direction	Description
reconfig_pdp_clk	Clock	Input	Reconfig Interface Clock. Intel recommends a frequency of 100 to 250 MHz for this clock.
reconfig_pdp_reset	reconfig_pdp_clk	Input	Reconfiguration Interface Reset
reconfig_pdp_address[13+K_d:0]	reconfig_pdp_clk	Input	Reconfig Interface Address. Word address. EMIB core adapter and soft CSR registers use unused space of F-tile Datapath Avalon memory mapped 16-bit address. Refer to Number of Datapath Memory Mapped Avalon Interfaces and Additional Address Bits per Interface for K_d values.
reconfig_pdp_byteenable [3:0]	reconfig_pdp_clk	Input	Byte Enable. If byteenable[3:0] is 4'b1111, 32-bit Dword Access is assumed; otherwise byte access is used.
reconfig_pdp_write	reconfig_pdp_clk	Input	Reconfig Write
reconfig_pdp_read	reconfig_pdp_clk	Input	Reconfig Read
reconfig_pdp_writedata [31:0]	reconfig_pdp_clk	Input	Reconfig Writedata
reconfig_pdp_readdata [31:0]	reconfig_pdp_clk	Output	Reconfig Read data
reconfig_pdp_waitrequest	reconfig_pdp_clk	Output	Reconfig Wait Request
reconfig_pdp_readdatavalid	reconfig_pdp_clk	Output	Reconfig Read Data Valid. Optional port, available if the port is enabled in parameter editor.

Table 50. Datapath Avalon Memory Mapped Interface Signals (Enable Separate Avalon Interface per fracture = 1)

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

Signal Name	Clocks Domain/Resets	Direction	Description
reconfig_pdp<*n*>_clk	Clock	Input	Reconfig Interface Clock. Intel recommends a frequency of 100 to 250 MHz for this clock.
reconfig_pdp<*n*>_reset_st<n>	reconfig_pdp<*n*>_clk	Input	Reconfig Interface Reset

continued...

Signal Name	Clocks Domain/Resets	Direction	Description
reconfig_pdp<*n*>_address[13:0]	reconfig_pdp<*n*>_clk	Input	Reconfig Interface Address. EMIB core adapter and soft CSR registers use unused space of F-tile Datapath. Avalon memory mapped 16-bit address. Soft CSR is only at _pdp0_ interface for single system IP.
reconfig_pdp<*n*>_byteenable [3:0]	reconfig_pdp<*n*>_clk	Input	Byte Enable. If byteenable[3:0] is 4'b1111, uses 32-bit Dword Access; otherwise uses byte access.
reconfig_pdp<*n*>_write_st	reconfig_pdp<*n*>_clk	Input	Reconfig Write
reconfig_pdp<*n*>_read_st	reconfig_pdp<*n*>_clk	Input	Reconfig Read
reconfig_pdp<*n*>_writedata[31:0]	reconfig_pdp<*n*>_clk	Input	Reconfig Writedata
reconfig_pdp<*n*>_readdata[31:0]	reconfig_pdp<*n*>_clk	Output	Reconfig Read data
reconfig_pdp<*n*>_waitrequest	reconfig_pdp<*n*>_clk	Output	Reconfig Wait Request
reconfig_pdp<*n*>_readdatavalid	reconfig_pdp<*n*>_clk	Output	Reconfig Read Data Valid. Optional port, available if the port is enabled on the GUI.

3.4.10.1. Number of Datapath Memory Mapped Avalon Interfaces and Additional Address Bits per Interface

Table 51. Number of Datapath Memory Mapped Avalon Interfaces and Additional Address Bits per Interface

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

FEC Enabled	Split Interface Enabled[28]	Number of Avalon interfaces	Additional Address Bits for Decoding (K_d)
0	0	1	K_d=Ceiling(log2(N))
0	1	N	K_d=0
1	X	1	K_d=0

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:[29]

Total tx_parallel_data or rx_parallel_data Bit Width Equation:

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

[28] Split Interface for datapath Memory mapped Avalon interface only supported for PMA Direct mode.

[29] This section explains the bit mapping of TX and RX parallel data if the **Provide separate interface for each PMA** option is disabled. If the **Provide separate interface for each PMA** option is enabled, refer to the introduction of Signal and Port Reference on page 110 to view the bit mapping differences.

Where:

- *N* = Number of PMA lanes value from 1 to 16.
- *X* = Number of streams for the PMA configuration. Depending on **PMA width**, *X* can be 1, 2, or 4.

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

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

Example 2: Total tx/rx_parallel_data Bit Width with 4 PMA Lanes (*N*=4) and 64-bit PMA Width (*X*=2)

```
tx_parallel_data [(80*4*2)-1:0] = tx_parallel_data [639:0]
rx_parallel_data [(80*4*2)-1:0] = rx_parallel_data [639:0]
```

Parallel Data Mapping information for TX and RX

If the **PMA width** is less than or equal to 32, *D*=**PMA width**.

If the **PMA width** is 64 or 128, *D*=32.

The lower case *x* is defined as *x*=`0 to` *X*`-1`. For a given lane, *n* and given stream *x*, you can calculate the TX and RX parallel data information according to the following tables:

Table 52. PMA Direct Mode TX Parallel Data Information Calculations (Enable Double width transfer = 1)

TX Parallel Data	MSB	LSB
Write Enable for TX Core FIFO in Elastic Mode (30)	79 + (80 * *x*) +(80 **n* * *X*)	
TX Data (Upper Data bits)	(40 + *D*-1) + (80 * *x*) + (80 **n* * *X*)	40 + (80 * *x*) + (80 **n* * *X*)
TX PMA Interface Data Valid Bit (31) (32)	38 + (80 * *x*) + (80 **n* * *X*)	
TX Data (Lower Data bits)	*D*-1 + (80 * *x*) + (80 **n* * *X*)	0 + (80 * *x*) +(80 **n* * *X*)

(30) Applicable only when using PMA clocking mode only and when TX/RX core FIFO is in elastic mode.

(31) Applicable only when using System PLL clocking mode.

(32) For all bonded configurations, all TX PMA Interface Data Valid bits must be asserted at the same cycle of `tx_coreclkin` clock.

Table 53. PMA Direct Mode RX Parallel Data Information Calculations (Enable Double width transfer = 1)

RX Parallel Data	MSB	LSB
Data valid for RX Core FIFO in Elastic Mode [25]	79 + (80 * x) + (80 *n * X)	
RX Deskew [33]	78 + (80 * x) + (80 *n * X)	
RX Data (Upper Data bits)	(40 + D-1) + (80 * x) + (80 *n * X)	40 + (80 * x) + (80 *n * X)
RX PMA Interface Data Valid Bit [26]	38 + (80 * x) + (80 *n * X)	
RX Data (Lower Data bits)	D-1 + (80 * x) + (80 *n * X)	0 + (80 * x) + (80 *n * X)

Table 54. PMA Direct Mode TX Parallel Data Information Calculations (Enable Double width transfer = 0)

TX Parallel Data	MSB	LSB
Write Enable for TX Core FIFO in Elastic Mode [25]	79 + (80 *n)	
TX PMA Interface Data Valid Bit [26] [27]	38 + (80 *n)	
TX Data	D-1 + (80 *n)	0 + (80 *n)

Table 55. PMA Direct Mode RX Parallel Data Information Calculations (Enable Double width transfer = 0)

RX Parallel Data	MSB	LSB
Data valid for RX Core FIFO in Elastic Mode [25]	79 + (80 *n)	
RX PMA Interface Data Valid Bit [26]	38 + (80 *n)	
RX Data	D-1 + (80 *n)	0 + (80 *n)

Table 56. FEC Direct Mode TX Parallel Data Information Calculations (Enable Double width transfer = 1)

TX Parallel Data	MSB	LSB
Alignment Marker[34]	77 + (80 * x) +(80 *n * X)	
TX Data (Upper 33 bits)	72 + (80 * x) + (80 *n * X)	40 + (80 * x) + (80 *n * X)
TX PMA Interface Data Valid Bit [26] [27]	38 + (80 * x) + (80 *n * X)	
Alignment Marker [25]	37 + (80 * x) + (80 *n * X)	
TX Data (Lower 31 bits)	32 + (80 * x) + (80 *n * X)	2 + (80 * x) + (80 *n * X)
Sync Head	1 + (80 * x) + (80 *n * X)	0 + (80 * x) + (80 *n * X)

[33] Applicable only when using PAM4 and X=2 or 4

[34] The two alignment markers in this table must be driven together by the same signal.

Table 57. FEC Direct Mode RX Parallel Data Information Calculations (Enable Double width transfer = 1)

RX Parallel Data	MSB	LSB
RX Deskew [35]	78 + (80 * x) + (80 *n * X)	
RX Data (Upper 33 bits)	72 + (80 * x) + (80 *n * X)	40 + (80 * x) + (80 *n * X)
RX PMA Interface Data Valid Bit [26] [36]	38	
Alignment Marker [29]	37	
RX Data (Lower 31 bits)	32 + (80 * x) + (80 *n * X)	2 + (80 * x) + (80 *n * X)
Sync Head	1 + (80 * x) + (80 *n * X)	0 + (80 * x) + (80 *n * X)

3.5.1. Parallel Data Mapping Information

Refer to the following references for parallel data mapping information for different PMA width combinations using the formulas specified in Bit Mapping for PMA and FEC Mode PHY TX and RX Datapath on page 121 .

- PMA Direct Mode Valid Parallel Data Bits
- FEC Direct Mode Valid Parallel Data Bits
- TX and RX Parallel Data Mapping Information for Different Configurations on page 126
- Example of TX Parallel Data for PMA Width = 8, 10, 16, 20, 32 (X=1) on page 131
- Example of TX Parallel Data for PMA width = 64 (X=2) on page 131
- Example of TX Parallel Data for PMA width = 64 (X=2) for FEC Direct Mode on page 132

Table 58. PMA Direct Mode Valid Parallel Data Bits

TX/ RX PMA Width	Enable TX/RX Double Width Transfer	Valid Parallel Data	Note
8	No	Data [7:0]	NA
10	No	Data [9:0]	NA
16	No	Data [15:0]	NA
20	No	Data [19:0]	NA
32	No	Data [31:0]	NA
8	Yes	Data [47:40] Data [7:0]	NA
10	Yes	Data [49:40]	NA

continued...

[35] Applicable only when using NRZ/PAM4 when X=2 or 4 or $N > 1$

[36] Only one per system

TX/ RX PMA Width	Enable TX/RX Double Width Transfer	Valid Parallel Data	Note
		Data [9:0]	
16	Yes	Data [55:40] Data [15:0]	Data [15:0] is the lower bits data. Data [55:40] is the upper bits data.
20	Yes	Data [59:40] Data [19:0]	Data [19:0] is the lower bits data. Data [59:40] is the upper bits data.
32	Yes	Data [71:40] Data [31:0]	Data [31:0] is the lower bits data. Data [71:40] is the upper bits data.
64	Yes	Data [151:120] Data [111:80] Data [71:40] Data [31:0]	Data [71:40] and Data [31:0] are the first stream data group. In this group, Data [31:0] is the lower bits data. Data [71:40] is the upper bits data. Data [151:120] and Data [111:80] are the second stream data group. Data [111:80] is the lower bits data. Data [151:120] is the upper bits data.
128	Yes	Data [311:280] Data [271:240] Data [231:200] Data [191:160] Data [151:120] Data [111:80] Data [71:40] Data [31:0]	Data [71:40] and Data [31:0] are the first stream data group. In this group, Data [31:0] is the lower bits data. Data [71:40] is the upper bits data. Data [151:120] and Data [111:80] are the second stream data group. In this group, Data [111:80] is the lower bits data. Data [151:120] is the upper bits data. Data [231:200] and Data [191:160] are the third stream data group. In this group, Data [191:160] is the lower bits data. Data [231:200] is the upper bits data. Data [311:280] and Data [271:240] are the fourth stream data group. In this group, Data [271:240] is the lower bits data. Data [311:280] is the upper bits data.

Table 59. FEC Direct Mode Valid Parallel Data Bits

TX/ RX PMA Width	Enable TX/RX Double Width Transfer	Valid Parallel Data	Note
32	Yes	Data [72:40] Data [32:2] Sync head [1:0]	Data [72:40] is the upper 33 bits data. Data [32:2] is the lower 31 bits data. Data [1:0] is the sync head.
64	Yes	Data [152:120] Data [112:82] Sync head [81:80] Data [72:40] Data [32:2] Sync head [1:0]	Second stream: Data [152:120] is the upper 33 bits data. Data [112:82] is the lower 31 bits data. Data [81:80] is the sync head First stream: Data [72:40] is the upper 33 bits data. Data [32:2] is the lower 31 bits data. Data [1:0] is the sync head
128	Yes	Data [312:280] Data [272:242] Sync head [241:240] Data [232:200]	Fourth stream: Data [312:280] is the upper 33 bits data. Data [272:242] is the lower 31 bits data. Data [241:240] is the sync head

continued...

TX/ RX PMA Width	Enable TX/RX Double Width Transfer	Valid Parallel Data	Note
		Data [192:162] Sync head [161:160] Data [152:120] Data [112:82] Sync head [81:80] Data [72:40] Data [32:2] Sync head [1:0]	Third stream: Data [232:200] is the upper 33 bits data. Data [192:162] is the lower 31 bits data. Data [161:160] is the sync head Second stream: Data [152:120] is the upper 33 bits data. Data [112:82] is the lower 31 bits data. Data [81:80] is the sync head First stream: Data [72:40] is the upper 33 bits data. Data [32:2] is the lower 31 bits data. Data [1:0] is the sync head

3.5.2. TX and RX Parallel Data Mapping Information for Different Configurations

The following show the TX and RX parallel data mapping information for different configurations, using the calculations from Bit Mapping for PMA and FEC Mode PHY TX and RX Datapath. Refer to Variables Defining Bits for the Interfacing Ports in Port and Signal Reference for full variable definitions.

Table 60. TX and RX Parallel Data Mapping Information (PMA Lanes, *N* = 1)

Table Notes:

1. For unused bits, you can tie the signal to '0' or '1' or leave it unconnected.
2. The core interface FIFO TX Write Enable and RX Data Valid signal is only valid when you are using PMA clocking mode, and when the core FIFO is in elastic mode.
3. The TX and RX PMA interface data valid signal is only valid when you are using System PLL clocking mode.

PMA Configuration	Bits	TX Parallel Data	RX Parallel Data
FGT PMA Width = 8, 10, 16, 20, 32 Single Width **(One PMA Lane [*N*=1] with PMA Width ≤ 32)** FHT PMA Width = 32 Single Width	79	Write Enable for TX Core FIFO in Elastic Mode	Data valid for RX Core FIFO in Elastic Mode
	38	TX PMA Interface Data Valid	RX PMA Interface Data Valid
	[*D*-1]:0	TX Data	RX Data
FGT PMA Width = 8, 10, 16, 20, 32 Double Width **(One PMA Lane [*N*=1] with PMA Width ≤ 32)** FHT PMA Width = 32 Double Width	79	Write Enable for TX Core FIFO in Elastic Mode	Data Valid for RX Core FIFO in Elastic Mode
	[*D* -1 + 40]:40	TX Data (Upper Data Bits)	RX Data (Upper Data Bits)
	38	TX PMA Interface Data Valid	RX PMA Interface Data Valid
	[*D* -1]:0	TX Data (Lower Data Bits)	RX Data (Lower Data Bits)
FGT/FHT	159	Write Enable for TX Core FIFO in Elastic Mode	Data Valid for RX Core FIFO in Elastic Mode

continued...

PMA Configuration	Bits	TX Parallel Data	RX Parallel Data
PMA Width = 64 Double Width **(One PMA Lane [*N*=1] with PMA Width = 64)** Second Stream	158		RX Deskew
	151:120	TX Data (Upper Data Bits)	RX Data (Upper Data Bits)
	118	TX PMA Interface Data Valid	RX PMA Interface Data Valid
	111:80	TX Data (Lower Data Bits)	RX Data (Lower Data Bits)
First Stream	79	Write Enable for TX Core FIFO in Elastic Mode	Data Valid for RX Core FIFO in Elastic Mode
	78		RX Deskew
	71:40	TX Data (Upper Data Bits)	RX Data (Upper Data Bits)
	38	TX PMA Interface Data Valid	RX PMA Interface Data Valid
	31:0	TX Data (Lower Data Bits)	RX Data (Lower Data Bits)
FHT PMA Width = 128 Double Width **(One PMA Lane [*N*=1] with PMA Width = 128)** Fourth Stream	319	Write Enable for TX Core FIFO in Elastic Mode	Data valid for RX Core FIFO in Elastic Mode
	318		RX Deskew
	311:280	TX Data (Upper Data Bits)	RX Data (Upper Data Bits)
	278	TX PMA Interface Data Valid	RX PMA Interface Data Valid
	271:240	TX Data (Lower Data Bits)	RX Data (Lower Data Bits)
Third Stream	239	Write Enable for TX Core FIFO in Elastic Mode	Data valid for RX Core FIFO in Elastic Mode
	238		RX Deskew
	231:200	TX Data (Upper Data Bits)	RX Data (Upper Data Bits)
	198	TX PMA Interface Data Valid	RX PMA Interface Data Valid
	191:160	TX Data (Lower Data Bits)	RX Data (Lower Data Bits)
Second Stream	159	Write Enable for TX Core FIFO in Elastic Mode	Data valid for RX Core FIFO in Elastic Mode
	158		RX Deskew
	151:120	TX Data (Upper Data Bits)	RX Data (Upper Data Bits)
	118	TX PMA Interface Data Valid	RX PMA Interface Data Valid
	111:80	TX Data (Lower Data Bits)	RX Data (Lower Data Bits)
First Stream	79	Write Enable for TX Core FIFO in Elastic Mode	Data valid for RX Core FIFO in Elastic Mode
	78		RX Deskew
	71:40	TX Data (Upper Data Bits)	RX Data (Upper Data Bits)
	38	TX PMA Interface Data Valid	RX PMA Interface Data Valid
	31:0	TX Data (Lower Data Bits)	RX Data (Lower Data Bits)

Table 61. TX and RX Parallel Data Mapping Information (Number of PMA Lanes (N) = 1)

PMA Configuration	Bits	TX Parallel Data	RX Parallel Data
FEC **FGT/FHT**	77	Alignment Marker	-

continued...

PMA Configuration	Bits	TX Parallel Data	RX Parallel Data
One PMA Lane (*N*)=1 Total Streams = 1 PMA Width = 32 First stream	72:40	TX Data (Upper 33 bits)	RX Data (Upper 33 Bits)
	38	TX PMA Interface Data Valid Bit	RX PMA Interface Data Valid Bit
	37	Alignment Marker	Alignment Marker
	32:2	TX Data (Lower 31 Bits)	RX Data (Lower 31 Bits)
	1:0	Sync Head	Sync Head
FEC FGT/FHT One PMA Lane (*N*) = 1 Total Streams = 2 PMA Width = 64 Second Stream	158	-	RX Deskew Bit
	157	Alignment Marker	-
	152:120	TX Data (Upper 33 bits)	RX Data (Upper 33 Bits)
	118	TX PMA Interface Data Valid Bit	-
	117	Alignment Marker	-
	112:82	TX Data (lower 31 bits)	RX Data (lower 31 bits)
	81:80	Sync Head	Sync Head
First stream	78	-	RX Deskew Bit
	77	Alignment Marker	-
	72:40	TX Data (Upper 33 Bits)	RX Data (Upper 33 Bits)
	38	TX PMA Interface Data Valid Bit	RX PMA Interface Data Valid Bit
	37	Alignment Marker	Alignment Marker
	32:2	TX Data (Lower 31 Bits)	RX Data (Lower 31 Bits)
	1:0	Sync Head	Sync Head
FEC FHT One PMA Lane(N) =1 Total Streams = 4 PMA Width = 128 Fourth Stream	318	-	RX Deskew Bit
	317	Alignment marker	-
	312:280	TX Data (Upper 33 Bits)	RX Data (Upper 33 Bits)
	278	TX PMA Interface Data Valid Bit	-
	277	Alignment Marker	-
	272:242	TX Data (Lower 31 Bits)	RX Data (Lower 31 Bits)
	241:240	Sync Head	Sync Head
Third Stream	238	-	RX Deskew Bit
	237	Alignment Marker	-
	232:200	TX Data (Upper 33 Bits)	RX Data (Upper 33 Bits)
	198	TX PMA Interface Data Valid Bit	-
	197	Alignment Marker	-
	192:162	TX Data (Lower 31 Bits)	RX Data (Lower 31 Bits)
	161:160	Sync Head	Sync Head
Second Stream	158		RX Deskew Bit
	157	Alignment Marker	-
	152:120	TX Data (Upper 33 Bits)	RX Data (upper 33 Bits)

continued...

PMA Configuration	Bits	TX Parallel Data	RX Parallel Data
	118	TX PMA Interface Data Valid Bit	-
	117	Alignment Marker	-
	112:82	TX Data (Lower 31 Bits)	RX Data (Lower 31 Bits)
	81:80	Sync Head	Sync Head
First stream	78	-	RX Deskew Bit
	77	Alignment Marker	-
	72:40	TX Data (Upper 33 Bits)	RX Data (Upper 33 Bits)
	38	TX PMA Interface Data Valid Bit	RX PMA Interface Data Valid Bit
	37	Alignment Marker	Alignment Marker
	32:2	TX Data (Lower 31 Bits)	RX Data (Lower 31 Bits)
	1:0	Sync Head	Sync Head

3.5.2.1. TX Parallel Data Mapping Information for SATA and USB Protocol Modes for Different Configurations

Note: For the SATA and USB protocol modes, support for the following features is preliminary in both simulation and hardware:

- PMA electrical idle feature
- Squelch detect feature
- Signal detect feature

The final support for these features is planned in a future version of the Intel Quartus Prime Pro Edition software.

When SATA or USB mode is selected for one PMA lane, the `tx_parallel_data` bus width is reduced from a 80-bit interface to a 76-bit interface. The formula outlined in Bit Mapping for PMA and FEC Mode PHY TX and RX Datapath to calculate the total `tx_parallel_data` bus is valid except the interface bus width is now reduced to 76-bit from 80-bit. The upper 4 bits are configured to `fgt_tx_pma_elecidle` bus signal. There is no change to the `rx_parallel_data` bus width.

Total tx_parallel_data Bit Width Equation for SATA and USB:

```
tx_parallel_data[(76*N)-1:0]
```

```
fgt_tx_pma_elecidle [(4*N)-1:0]
```

Where:

- *N* = Number of PMA lanes value from 1 to 16.

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

Example 1: Total tx_parallel_data Bit Width with 1 SATA Link (*N*=1) and 32-bit PMA Width

```
tx_parallel_data [(76*1)-1:0] = tx_parallel_data [75:0], fgt_tx_pma_elecidle
[3:0]
```

Example 2: Total tx_parallel_data Bit Width with 2 SATA Links (*N*=2) and 32-bit PMA Width

```
tx_parallel_data [(76*2)-1:0] = tx_parallel_data [151:0], fgt_tx_pma_elecidle
[7:0]
```

The following table shows the TX parallel data mapping information for the SATA and USB protocol modes for different configurations.

Table 62. TX Parallel Data Mapping Information for SATA and USB Protocol Modes (PMA Lanes, *N* = 1)

Table Notes:

1. For unused bits, you can tie the signal to '0' or '1' or leave it unconnected.
2. The core interface FIFO TX Write Enable and RX Data Valid signal is only valid when you are using PMA clocking mode, and when the core FIFO is in elastic mode.
3. The TX and RX PMA interface data valid signal is only valid when you are using System PLL clocking mode.
4. The table is applicable when you select **Enable simplified TX data interface**.

Note: For every PMA lane count (*N*) increase, add a 76-bit offset to the defined bits in the table below. For example if configuration is: PMA Width = 32, Single Width, SATA and USB, *N*=2 then Write Enable for TX Core FIFO in Elastic Mode = bit[151], TX PMA Interface Data Valid = bit[111] and TX data = bits[107:76].

PMA Configuration	Bits	TX Parallel Data	RX Parallel Data
FGT PMA Width = 8, 10, 16, 20, 32 Single Width SATA and USB **(One PMA Lane [*N*=1] with PMA Width ≤ 32)**	75	Write Enable for TX Core FIFO in Elastic Mode	No change, refer to TX and RX Parallel Data Mapping Information (PMA Lanes, *N* = 1) with the same PMA configuration.
	35	TX PMA Interface Data Valid	
	[*D*-1]:0	TX Data	
FGT PMA Width = 8, 10, 16, 20, 32 Double Width SATA and USB **(One PMA Lane [*N*=1] with PMA Width ≤ 32)**	75	Write Enable for TX Core FIFO in Elastic Mode	No change, refer to TX and RX Parallel Data Mapping Information (PMA Lanes, *N* = 1) with the same PMA configuration.
	[*D* -1 + 36]:36	TX Data (Upper Data Bits)	
	35	TX PMA Interface Data Valid	
	[*D* -1]:0	TX Data (Lower Data Bits)	
RS-FEC Enabled, *N* = number of FEC lanes, *X* = fec stream index = (0:*N*-1)			
FGT PMA Width = 32 Double Width SATA and USB **(One PMA Lane [*N*=1] with PMA width = 32)**	74	TX Deskew Bit	No change, refer to TX and RX Parallel Data Mapping Information (Number of PMA Lanes (N) = 1) with the same PMA configuration.
	73	TX Alignment Marker	
	68:36	TX Data (Upper 33 Bits)	
	35	TX PMA Interface Data Valid Bit	
	32:2	TX Data (Lower 31 Bits)	
	1:0	Sync head	

3.5.3. Example of TX Parallel Data for PMA Width = 8, 10, 16, 20, 32 (X=1)

The following data is specific to the *X*=1 case. *N* indicates the number of PMA lanes. For a given *N*, *n* can be from 0 --> *N*-1. *N* can be up to 16 for FGT, and up to 4 for FHT, and depends on the number of PMA lanes and PMA width configuration. Enable Double width transfer = 0. Refer to Variables Defining Bits for the Interfacing Ports in Port and Signal Reference for full variable definitions.

Table 63. Example of TX Parallel Data bits for PMA Width = 8, 10, 16, 20, 32 (*X*=1)

Bits	TX Parallel Data for *n*=0	Bits	TX Parallel Data for *n*=1	••	Bits	TX Parallel Data for *n*=15
79	Write Enable for TX Core FIFO in Elastic Mode	159	Write Enable for TX Core FIFO in Elastic Mode	•••	1279	Write Enable for TX Core FIFO in Elastic Mode
38	TX PMA Interface Data Valid	118	TX PMA Interface Data Valid		1238	TX PMA Interface Data Valid
31:0	TX Data	111:80	TX Data		1231:1200	TX Data

The following are the TX PMA Interface Data Valid signals for each of the PMA Lanes in Example of TX Parallel Data bits for PMA Width = 8, 10, 16, 20, 32 (*X*=1):

- If *N*=1, `tx_parallel_data [38]`
- If *N*=2, `tx_parallel_data [118]`

..

- If *N*=16, `tx_parallel_data [1238]`

3.5.4. Example of TX Parallel Data for PMA width = 64 (X=2)

The following data is specific to the *X*=2 case. *N* indicates the number of PMA lanes. For a given *N*, *n* can be from 0 --> *N*-1. *N* can be up to 8 for FGT, and up to 4 for FHT, and depends on the number of PMA lanes. Enable Double width transfer = 1. Refer to Variables Defining Bits for the Interfacing Ports in Port and Signal Reference for full variable definitions.

Table 64. Example of TX Parallel Data for PMA width = 64 (X =2)

Condition	Bits	TX Parallel Data for *n*=0	Bits	TX Parallel Data for *n*=1	••	Bits	TX Parallel Data for *n*=7
Second Stream	159	Write Enable for TX Core FIFO in Elastic Mode	319	Write Enable for TX Core FIFO in Elastic Mode	••	1279	Write Enable for TX Core FIFO in Elastic Mode
	151:120	TX Data (Upper Data Bits)	311:280	TX Data (Upper Data Bits)		1391:1240	TX Data (Upper Data Bits)
	118	TX PMA Interface Data Valid	278	TX PMA Interface Data Valid		1238	TX PMA Interface Data Valid

continued...

Condition	Bits	TX Parallel Data for *n*=0	Bits	TX Parallel Data for *n*=1	••	Bits	TX Parallel Data for *n*=7
	111:80	TX Data (Lower Data Bits)	271:240	TX Data (Lower Data Bits)		1231:1200	TX Data (Lower Data Bits)
First Stream	79	Write Enable for TX Core FIFO in Elastic Mode	239	Write Enable for TX Core FIFO in Elastic Mode	••	1199	Write Enable for TX Core FIFO in Elastic Mode
	71:40	TX Data (Upper Data Bits)	231:200	TX Data (Upper Data Bits)		1191:1160	TX Data (Upper Data Bits)
	38	TX PMA Interface Data Valid	198	TX PMA Interface Data Valid	••	1158	TX PMA Interface Data Valid
	31:0	TX Data (Lower Data Bits)	191:160	TX Data (Lower Data Bits)		1151:1120	TX Data (Lower Data Bits)

The following are the TX PMA Interface Data Valid signals for each of PMA Lanes, for each stream, in Example of TX Parallel Data for PMA width = 64 (X =2).

- If PMA lanes *n*=0, for first stream data, data valid signal = `tx_parallel_data [38]`
- If PMA lanes *n*=0, for second stream data, data valid signal = `tx_parallel_data [118]`
- If PMA lanes *n*=1, for first stream data, data valid signal = `tx_parallel_data [198]`
- If PMA lanes *n*=1, for second stream data, data valid signal = `tx_parallel_data [278]`

3.5.5. Example of TX Parallel Data for PMA width = 64 (X=2) for FEC Direct Mode

The following data is specific to the *X*=2 case for a FEC Direct mode. *N* indicates the number of PMA lanes. For a given *N*, *n* can be from 0 --> *N*-1. *N* can be up to 8 for FGT, and up to 4 for FHT, and depends on the number of PMA lanes. Enable Double width transfer Enabled = 1. Refer to Variables Defining Bits for the Interfacing Ports in Port and Signal Reference for full variable definitions.

Table 65. Example of TX Parallel Data for PMA width = 64 (*X*=2) for FEC Direct Mode

	Bits	TX Parallel Data for *n*-0	Bits	TX Parallel Data for *n*=1	..	Bits	TX Parallel Data for *n*=7
Second Stream	157	TX Alignment marker	317	TX Alignment marker		1277	TX Alignment Marker
	152:120	TX Data(Upper 33 bits)	312:280	TX Data (Upper 33 bits)		1272:1240	TX Data (Upper 33 bits)

continued...

	118	TX PMA Interface Data Valid Bit	278	TX PMA Interface Data Valid Bit	..	1238	TX PMA Interface Data Valid Bit
	117	TX Alignment Marker	277	TX Alignment Marker		1237	TX Alignment Marker
	112:82	TX Data (Lower 31 bits)	272:242	TX Data (lower 31 bits)		1232:1202	TX Data (Lower 31 bits)
	81:80	Sync Head	241:240	Sync Head		1201:1200	Sync Head
First Stream	77	TX Alignment Marker	237	TX Alignment Marker		1197	TX Alignment Marker
	72:40	TX Data (Upper 33 Bits)	232:200	TX Data(upper 33 bits)		1192:1160	TX Data (upper 33 bits)
	38	TX PMA Interface Data Valid Bit	198	TX PMA Interface Data Valid Bit	..	1158	TX PMA Interface Data Valid Bit
	37	TX Alignment Marker	197	TX Alignment Marker		1157	TX Alignment Marker
	32:2	TX Data (Lower 31 bits)	192:162	TX Data(Lower 31 bits)		1152:1122	TX Data(lower 31 bits)
	1:0	Sync Head	161:160	Sync Head		1121:1120	Sync Head

3.6. Clocking

F-tile supports six different clock output options from TX and RX that you can use for FPGA core clocking.

Note: Refer to Guidelines for F-Tile Reference and System PLL Clocks Intel FPGA IP Usage on page 195 for reference clock and system PLL usage.

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.

Bond Clock

The bond clock and the word clock are always PMA parallel clocks, and they are usually the same. However, in certain multi-lane (that is, more than one PMA) bonded configurations, the bond clock from every core interface comes from one primary PMA, while the word clock comes from the PMA associated with that core interface. Refer to Recommended Connection and Source for connection recommendations.

User Clock 1

User clock 1 is the divided version of the PMA data rate. The available division factor for user clock 1 depends on the PMA type.

FHT: The user clock 1 for FHT is calculated as the PMA data rate divided by (number of streams * division factor).

$$FHT\ User\ clock\ 1 = \frac{PMA\ datarate}{Number\ of\ streams\ per\ PMA * Division\ Factor}$$

You can specify a division factor using **FHT user clk div33_34 select** and **FHT TX/RX user clk1 select**. Division factors for FHT can be 33, 34, 66, 68, or 40.

FGT: The user clock 1 for FGT 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.

$$FGT\ User\ clock\ 1 = \frac{VCO\ frequency}{Division\ factor}$$

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

User Clock 2

User clock 2 is also a divided version of the PMA data rate.

FHT User clock 1 and 2:

In FHT, user clock 2 can be different from user clock 1 if you select a different division factor.

By default, user clock 1 and user clock 2 are disabled; you can enable either or both. The following table shows different combinations of User clock 1 and User clock 2 based on parameters selected in the parameter editor.

Table 66. Some Possible Combinations of FHT User clock 1 and User clock 2 Based on Parameters Selected in the Parameter Editor

FHT user clk div33_34 select	FHT TX/RX user clk1 select	FHT TX/RX user clk2 select	User clock 1	User clock 2
DIV_33	DIV_33_34	DIV_40	PMA Data rate / (Number of streams per PMA * 33)	PMA Data rate / (Number of streams per PMA *40)
DIV_33	DIV_40	DIV_33_34	PMA Data rate / (Number of streams per PMA *40)	PMA Data rate / (Number of streams per PMA * 33)
DIV_34	DIV_33_34	DIV_40	PMA Data rate / (Number of streams per PMA * 34)	PMA Data rate / (Number of streams per PMA * 40)
DIV_34	DIV_40	DIV_33_34	PMA Data rate / (Number of streams per PMA * 40)	PMA Data rate / (Number of streams per PMA * 34)
DIV_66	DIV_33_34	DIV_40	PMA Data rate / (Number of streams per PMA * 66)	PMA Data rate / (Number of streams per PMA * 40)
DIV_66	DIV_40	DIV_33_34	PMA Data rate / (Number of streams per PMA * 40)	PMA Data rate / (Number of streams per PMA * 66)
DIV_68	DIV_33_34	DIV_40	PMA Data rate / (Number of streams per PMA * 68)	PMA Data rate / (Number of streams per PMA * 40)
DIV_68	DIV_40	DIV_33_34	PMA Data rate / (Number of streams per PMA * 40)	PMA Data rate / (Number of streams per PMA * 68)

Figure 74. FHT TX User clock 1 and User clock 2

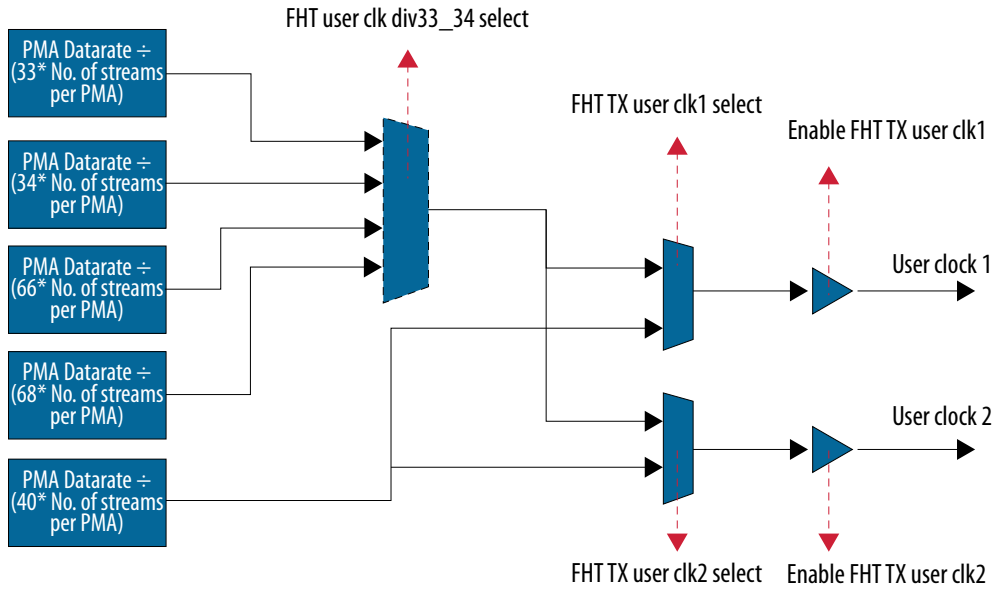
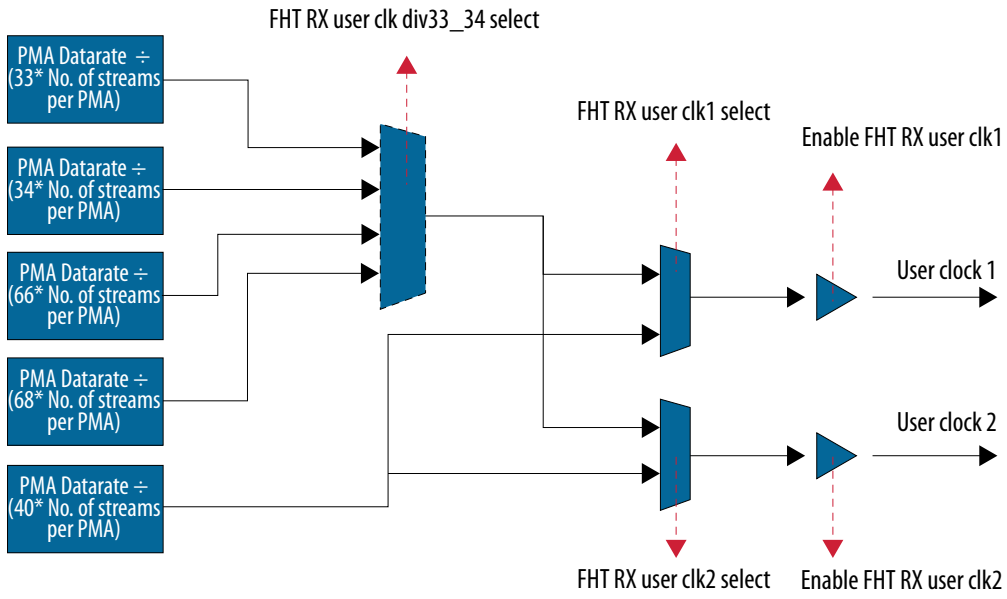


Figure 75. FHT RX User clock 1 and User clock 2



FGT User clock 1 and 2:

By default, user clock 1 and user clock 2 are disabled; you can enable either or both. In FGT, user clock 2 is same as user clock 1. The following table shows some example **TX/RX user clock div by** parameter values. Valid values can range from 12 to 139.5 in increments of 0.5.

Table 67. Example TX/RX user clock div by Parameter Values

TX/RX user clock div by	User Clock 1	User Clock 2
33	VCO Frequency ÷ 33	VCO Frequency ÷ 33
34	VCO Frequency ÷ 34	VCO Frequency ÷ 34
66	VCO Frequency ÷ 66	VCO Frequency ÷ 66
68	VCO Frequency ÷ 68	VCO Frequency ÷ 68

Figure 76. FGT TX User clock 1 and User clock 2

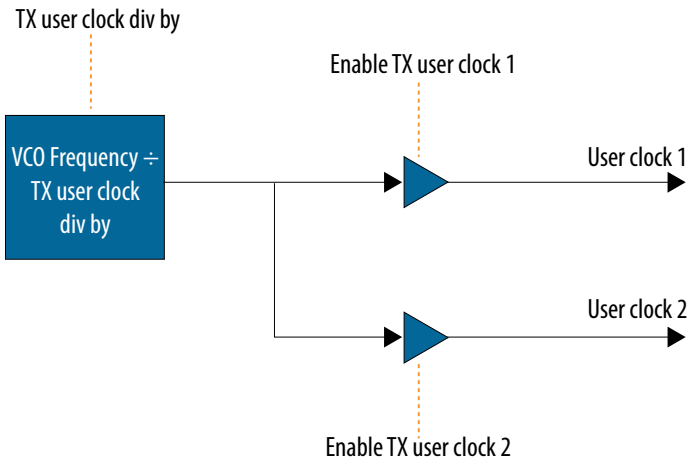
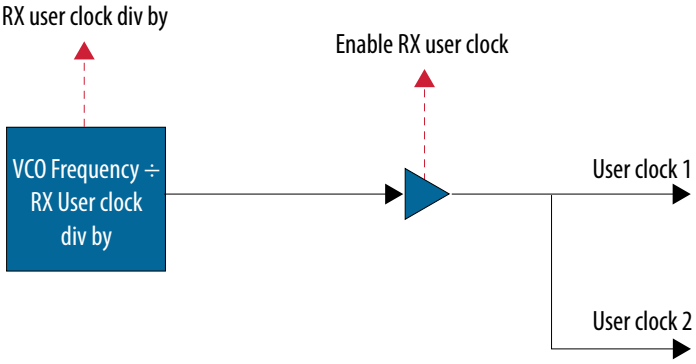


Figure 77. FGT RX User clock 1 and User clock 2



On the TX side of FGT, you can enable user clock 1 and user clock 2 separately. On the RX side, when you enable user clock, it enables both user clock 1 and user clock 2.

The TX and RX clocks for word clock, bond clock, user clock 1, and user clock 2, 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 F-Tile PMA/FEC Direct PHY Intel FPGA IP.

Sys PLL Clock Div 2

The Sys PLL clock Div 2 is the output clock from the system PLL, divided by 2. The frequency of this clock is same as the output frequency of the system PLL connected to the current instance of the F-Tile PMA/FEC Direct PHY Intel FPGA IP, divided by 2.

The Sys PLL clock and Sys PLL clock Div 2 for TX and RX are the same clock and are derived from one system PLL.

3.6.1. Clock Ports

The F-Tile PMA/FEC Direct PHY Intel FPGA IP supports two clock output ports.

The two clock output ports can each choose one of the six 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 six 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 six 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.

The difference between `tx/rx_clkout2` and `tx/rx_clkout` is that it can further divide the six clock options by a factor specified in the **tx/rx_clkout2 clock div by** menu.

Available `tx_clkout2` divide-by options are: 1, 2, 4. Available `rx_clkout2` divide-by options are: 1, 2.

Figure 78. tx_clkout and tx_clkout2

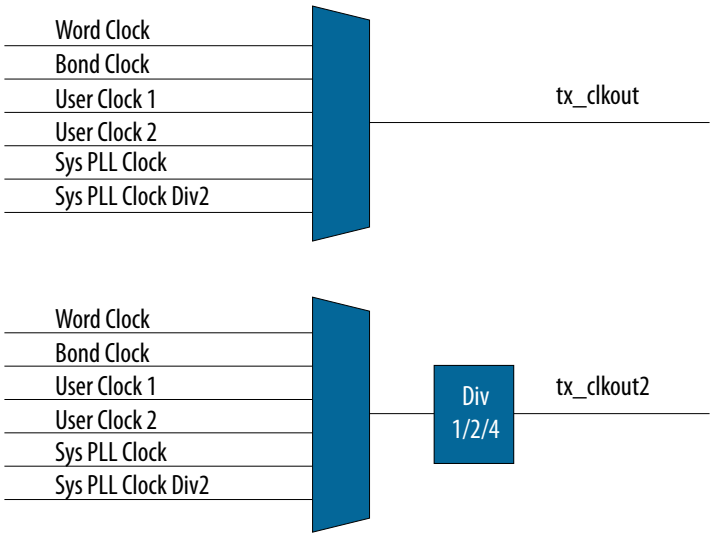
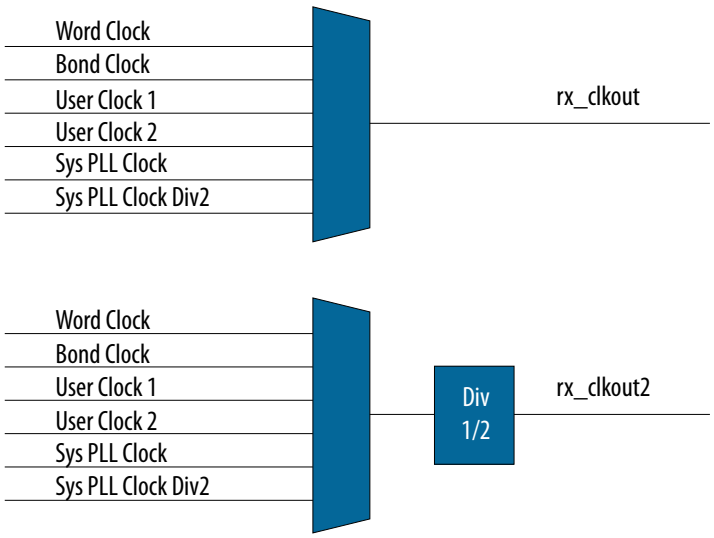


Figure 79. rx_clkout and rx_clkout2



When you select user clock 1 or user clock 2 as the source clock for `tx/rx_clkout` or `tx/rx_clkout2`, ensure that you also enable user clock 1 or user clock 2, as appropriate. If you are using FHT, you can enable user clock 1 or user clock 2 by enabling **Enable FHT TX/RX user clk1** or **Enable FHT TX/RX user clk2** in **TX/RX FHT PMA** on the **TX/RX Datapath Options** tab.

When using FGT, on the TX side you can enable user clock 1 or user clock 2 by enabling **TX User Clock Settings ➤ Enable TX user clock**.

tx/rx_coreclkin

`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 `tx/rx_clkout` and `tx/rx_clkout2` when connecting to `tx/rx_coreclkin` is shown in Recommended tx/rx_coreclkin Connection and tx/rx_clkout2 Source on page 139. The recommended port connections details are shown in Port Widths and Recommended Connections for tx/rx_coreclkin, tx/rx_clkout, and tx/rx_clkout2 on page 140.

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

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

Table 68. 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 if tx/rx_clkout2
PMA	PC	No	tx/rx_clkout	Word/Bond clock	N/A
		Yes	tx/rx_clkout2	Word/Bond clock	2
	Elastic	Yes	tx/rx_clkout2 or any other clock source from user	Word/Bond clock/ User clock1 or 2	2
		No	tx/rx_clkout or any other clock source from user	Word/Bond clock/ User clock1 or 2	N/A
System PLL	PC	No	tx/rx_clkout	Sys PLL clock	N/A
		Yes	tx/rx_clkout	Sys PLL clock Div2	N/A

- When using system PLL clocking mode, both `tx_clkout` and `rx_clkout` can clock `tx_coreclkin` and `rx_coreclkin`.
- When using PMA clocking mode, `tx_cllkout/2` must clock `tx_coreclkin`. `rx_clkout/2` must clock `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), `tx_clkout` or `rx_clkout` can clock both `tx_coreclkin` and `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:

- `tx_clkout` and `rx_clkout`
- `tx_clkout2` and `rx_clkout2`
- `tx_coreclkin` and `rx_coreclkin`

Table 69. Port Widths and Recommended Connections

Refer to Variables Defining Bits for the Interfacing Ports in Port and Signal Reference 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 `tx_clkout [0]` or `tx_clkout2[0]` to `tx_coreclkin[N-1:0]` • Connect `rx_clkout [0]` or `rx_clkout2[0]` to `rx_coreclkin[N-1: 0]`
64	2 * *N*	• Connect `tx_clkout [0]` or `tx_clkout2[0]` to `tx_coreclkin[2 * N-1 : 0]` • Connect `rx_clkout [0]` or `rx_clkout2[0]` to `rx_coreclkin[2*N-1: 0]`
128	4 * *N*	• Connect `tx_clkout [0]` or `tx_clkout2[0]` to `tx_coreclkin[4 * N-1 : 0]` • Connect `rx_clkout [0]` or `rx_clkout2[0]` to `rx_coreclkin[4*N-1: 0]`

Table 70. Example with Number of PMA Lane = 1

PMA Width	tx/rx_clkout/2 Port Width	Recommended Connection
8, 10, 16, 20, 32	1	• Connect tx_clkout or tx_clkout2 to tx_coreclkin • Connect rx_clkout or rx_clkout2 to rx_coreclkin
64	2	• Connect tx_clkout[0] or tx_clkout2[0] to tx_coreclkin[1:0] • Connect rx_clkout[0] or rx_clkout2[0] to rx_coreclkin[1:0]
128	4	• Connect tx_clkout[0] or tx_clkout2[0] to tx_coreclkin[3:0] • Connect rx_clkout[0] or rx_clkout2[0] to rx_coreclkin[3:0]

Table 71. Example with Number of PMA Lanes = 4

PMA Width	tx/ rx_clkout/2 Port Width	Recommended Connection
8, 10, 16, 20, 32	4	• Connect tx_clkout[0] or tx_clkout2[0] to tx_coreclkin[3:0] • Connect rx_clkout[0] or rx_clkout2[0] to rx_coreclkin[3:0]
64	8	• Connect tx_clkout[0] or tx_clkout2[0] to tx_coreclkin[7:0] • Connect rx_clkout[0] or rx_clkout2[0] to rx_coreclkin[7:0]
128	16	• Connect tx_clkout[0] or tx_clkout2[0] to tx_coreclkin[15:0] • Connect rx_clkout[0] or rx_clkout2[0] to rx_coreclkin[15:0]

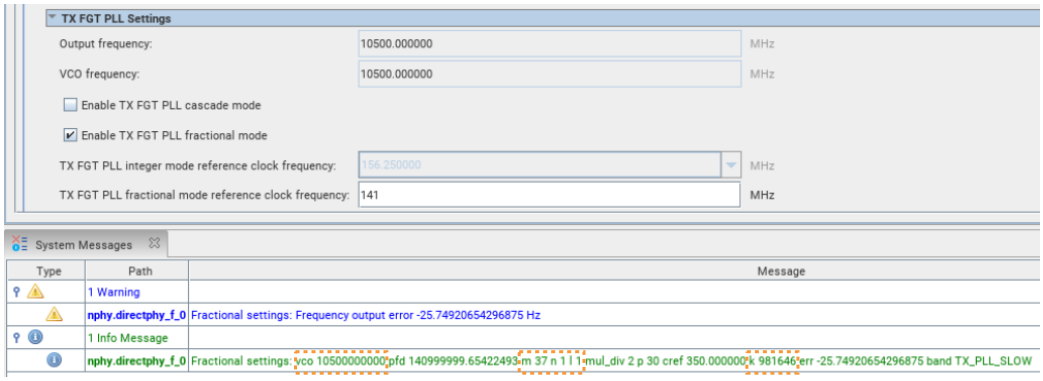
3.6.4. FGT PMA Fractional Mode

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.

- When enabling fractional mode, Intel recommends using 141 MHz reference clock frequency to maximize the distance on the PLL spurs across all OTN / SDI / CPRI rates.
- 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 following figure.
- For a given data rate, you must enable fractional mode if you need to dynamically configure the *k* counter during run time.

Figure 80. Example of Fractional Settings in System Messages Tab



FGT PMA supports fractional mode in following PMA modes:

Table 72. FGT PMA Support for Fractional Mode

PMA Mode	Fractional Mode Support
TX simplex	TX FGT PLL supports fractional mode in TX simplex. To enable, select **TX simplex** option for **PMA mode**, and turn on the **Enable TX FGT 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 16 FGT TX PMAs. *Note:* FGT PMA does not support fractional mode for RX simplex.
Duplex	FGT 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 **16** for the **Number of PMA lanes**, and turn on **Enable TX FGT 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 leave the `rx_cdr_refclk_link` port unconnected and it is tied to ground internally in the IP core. • You can place duplex fractional mode on any of 16 FGT PMAs.
Primary PLL configuration	To enable fractional mode with the primary PLL configuration, select the **Duplex** option for **PMA mode**, select **2** or **4** for the **Number of PMA lanes**, and turn on **Enable TX FGT PLL fractional mode** and **Enable TX FGT PLL cascade mode** options in the parameter editor. • In Primary PLL configuration, the TX PLL of one lane is in fractional mode and acts as the reference clock source for the local CDR and TX PLL and RX CDR blocks of other lanes (configured in integer mode) within the quad. • Primary PLL configuration is not supported when you select 6 ,8, 12, or 16 for the number of PMA lanes. • You can place the primary PLL configuration with 2 PMA lanes either on FGT PMA Lane 1 and 0 (same quad) of any quad, with Lane 1 as the primary. On FGT PMA Lane 3 and 2 (same quad) of any quad with lane 3 being the primary. • You can place the primary PLL configuration with 4 PMA lanes on FGT PMA Lane 3,2,1 and 0 (same quad) of any quad with Lane 3 being the primary. • When you place any primary PLL configuration in Quad 2, you cannot configure reference clock [8] (accessible by Quad2) as output to provide RX recovered clock. • When you place any primary PLL configuration in Quad 3, you cannot configure reference clock [9] (accessible by Quad3) as output to provide RX recovered clock.

Note: Refer to *FGT PLL Configuration* in *F-tile Architecture User Guide* for more information.

Tuning the k Counter Value in Fractional Mode

For designs with 1, 2, or 4 PMA lanes, you can configure the FGT PMA in fractional mode to adjust the frequency and datarate by a small amount for rate matching purposes.

- When number of PMA lanes is 1, turn on **Enable TX FGT PLL fractional mode**, set PMA reference clock frequency as 141 MHz, and tune the k counter value of the lane.
- When number of PMA lanes is 2, or 4, turn on **Enable TX FGT PLL fractional mode**, turn on **Enable TX FGT PLL cascade mode**, set PMA reference clock frequency as 141 MHz, and tune the k counter value of the primary lane.

In order to meet the jitter specifications of OTN (Optical Transport Network) and SDI (Serial Digital Interface), dynamic changes in the fractional value should have the following limits:

- Maximum step size: 2.5 ppm
- Minimum duration between steps: 1 us

If the OTN or SDI jitter specifications does not apply to your design, and you want to adjust the data rate without losing lock, then dynamic changes in the fractional value should have the following limits:

- Maximum step size: 100 ppm
- Minimum duration between steps: unknown

Intel recommends keeping the duration as long as you can afford to get stable performance.

Each FGT PMA has an Avalon memory-mapped interface register containing the *k* counter. The *k* counter / 2^22 gives the fractional value *K* of the feedback counter. The fractional value *K* plus the *M* counter value provides the total feedback counter and determines how much PPM each bit in the *k* counter represents. For example, the LSB (least significant bit) in the *k* counter represents PPM = (1 / 2^22) / (*K*+*M*) × (10^6).

The procedure to change the *k* counter is:

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

Each FGT PMA contains 3 PLLs; slow, medium and fast. FGT PMAs are organized in a quad. The *k* counter and strobe bit Avalon memory-mapped interface register addresses depend on the location of the transceiver in the quad and which PLL is used (slow, medium, fast) as shown in the table below.

Table 73. FGT PMA Fractional *k* Counter and Strobe Register Addresses

Channel Location in Quad	PLL	Fractional *k* Counter Register	Strobe Register
0	Slow	0x44000[30:9]	0x4400C[17]
	Medium	0x44100[30:9]	0x4410C[17]
	Fast	0x44200[30:9]	0x4420C[17]
1	Slow	0x4C000[30:9]	0x4C00C[17]

continued...

Channel Location in Quad	PLL	Fractional *k* Counter Register	Strobe Register
	Medium	0x4C100[30:9]	0x4C10C[17]
	Fast	0x4C200[30:9]	0x4C20C[17]
2	Slow	0x54000[30:9]	0x5r00C[17]
	Medium	0x54100[30:9]	0x5410C[17]
	Fast	0x54200[30:9]	0x5420C[17]
3	Slow	0x5C000[30:9]	0x5C00C[17]
	Medium	0x5C100[30:9]	0x5C10C[17]
	Fast	0x5C200[30:9]	0x5C20C[17]

You can find the transceiver location and PLL selected in the <design_name>.syn.rpt generated by the Intel Quartus Prime Pro Edition software as shown in the following figure.

Figure 81. Sample Intel Quartus Prime Pro Edition Software Synthesis Compilation Result

```
; z1577a_u_ux_quad_3__ux3_synth_lc_med_en         ;
enable                                    ; String        ;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_f_out_hz          ;
00000000010101100110110100111111010000000 ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_f_pfd_hz          ;
00000000000000000000000000000000000000000 ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_f_ref_hz          ;
00000000000010001101100111101111000100000 ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_f_rx_postdiv_hz    ;
00000000000010001010010010000110010000000 ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_f_tx_postdiv_hz    ;
00000000000001101110101000000101000000000 ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_f_vco_hz          ;
00000010101100110110100111111010000000000 ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_fractional_en      ;
enable                                    ; String        ;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_k_counter         ;
0000111010100111001110                    ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_l_counter         ;
001000                                    ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_m_counter         ;
000100111                                 ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_n_counter         ;
000001                                    ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_powerdown_mode     ;
false                                     ; String        ;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_primary_use        ;
ux3_synth_lc_med_primary_use_disabled     ; String        ;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_rx_postdiv_counter ;
00101000                                  ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_tx_postdiv_counter ;
01100100                                  ; Unsigned Binary;
; z1577a_u_ux_quad_3__ux3_synth_lc_med_tx_postdiv_fractional_en ;
disable                             ; String        ;
```

In the sample compilation result shown above, the FGT PMA is placed in quad 3, channel 3 and the medium PLL is used. The *M* counter is 39 (000100111) and the nominal *K* value is 0.057 (0000111010100111001110/2^22). The LSB in the *k* counter represents 6 ppb (parts per billion) (1/2^22/39.057). The *k* counter register address is 0x5C100[30:9] and the strobe register bit address is 0x5C10C[17].

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 Custom Cadence Generation Ports and Logic Use Cases 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 74. 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.(37)	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 `tx_cadence`, `tx_cadence_fast_clk`, and `tx_cadence_slow_clk` ports are available in the F-Tile PMA/FEC Direct PHY Intel FPGA IP. CCG logic uses the `tx_cadence_fast_clk` and `tx_cadence_slow_clk` inputs (does not monitor PMA Interface FIFO status), and generates a `tx_cadence` output signal. You must use `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 on page 124.

Table 75. tx_cadence_fast_clk and tx_cadence_slow_clk connections

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

(37) 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.

Related Information

- Enabling Custom Cadence Generation Ports and Logic on page 211
- TX PMA interface Parameters on page 95

3.7.1. Enabling the tx_cadence_slow_clk_locked Port

If the `tx_cadence_slow_clk` signal does not come directly from TX PLL (word clock, bond clock, 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 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 `tx_cadence_slow_clk_locked`.

3.7.2. Rate Match FIFO

When using the system PLL clocking mode, you must create and instantiate a rate match FIFO in PMA/FEC Direct mode when the user FPGA core logic (user clock domain) runs at a different frequency than system PLL frequency (system PLL frequency ÷ 2 when double width transfer is enabled). You must create and implement this rate match FIFO for the clock domain transfer from the user clock domain to the system PLL clock domain.

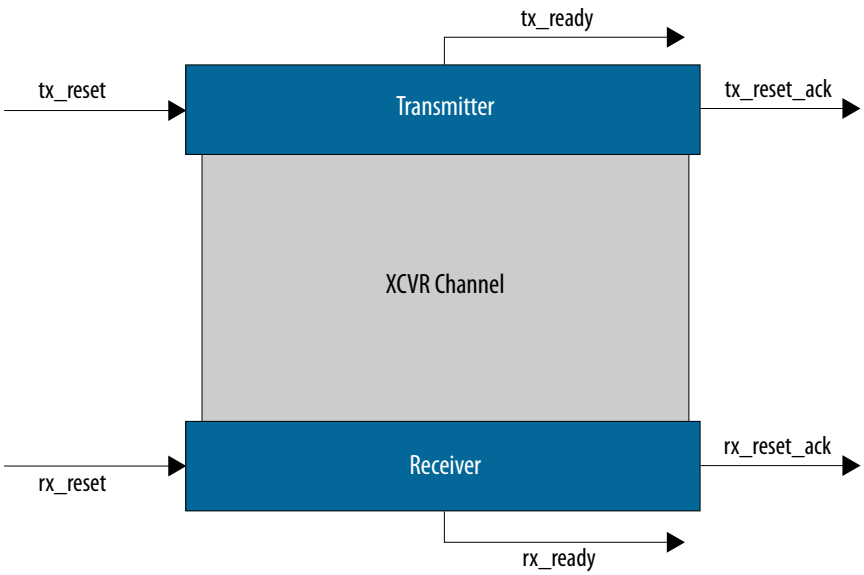
As the rate match FIFO is not available in IP catalog, you must create the FIFO. Implement the FIFO by placing a rate-matching soft FIFO between your logic and the core for pacing the data valid signal. Use this technique whenever the user FPGA core logic (user clock domain) runs at a different frequency than system PLL frequency (system PLL frequency ÷ 2 when double width transfer is enabled).

3.8. Asserting Reset

A soft reset controller handles all non-PCIe reset scheduling for F-tile in PMA Direct and FEC Direct modes. The controller flow is an internal component of the larger F-tile IP flow, and is supported by a tile IP auto-instantiation and auto-connection flow.

Asserting a reset sequence ensures that the physical medium attachment (PMA) in each channel initializes and functions correctly. You can reset the transmitter (TX) and receiver (RX) data paths independently or together.

Figure 82. Reset Diagram



3.8.1. Reset Signal Requirements

The following requirements apply to reset signal use for F-tile PMA/FEC Direct PHY designs:

- Ensure that `tx_reset`/`rx_reset` remain asserted until `tx_reset_ack`/`rx_reset_ack` goes high.
- Expect random data if `tx_ready`/`rx_ready` is not asserted.
- In forward error correction (FEC) modes, during reset sequencing, after `tx_am_gen_start` is asserted, start sending alignment markers and assert `tx_am_gen_2x_ack` after two alignment markers are sent. `tx_am_gen_start` goes high as part of reset sequence (that is, before `tx_ready` is asserted).
- In FEC modes when sending alignment markers, you can pace tx data valid with the `tx_cadence` signal.
- For duplex configurations, you can assert `tx_reset` and `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 76. Reset Signals—Block Level

Reset Signal	TX PHY	TX Datapath	RX PHY	RX Datapath	Soft CSRs	PMA Reconfiguration Interface
tx_reset	Yes	Yes	No	No	No	No
rx_reset	No	No	Yes	Yes	No	No
reconfig_xcvr_reset	No	No	No	No	No	Yes
reconfig_pdp_reset	No	No	No	No	Yes	No

Note: It is recommended that you assert the reconfig_xcvr_reset signal once upon power-up to ensure the PMA reconfiguration interface is placed into a known, good state.

3.8.4. Reset Signals—Descriptions

Table 77. Reset Signal Descriptions

Name	Width	Domain	Direction	Type	Description
tx_reset	1	Asynchronous	Input	N/A	TX reset input for TX PMAs and TX datapath. Must be kept asserted until tx_reset_ack is asserted. Applies to all TX channels in a F-Tile PMA/FEC Direct PHY Intel FPGA IP instance.
tx_reset_ack	1	Asynchronous	Output	N/A	TX fully in reset indicator. This signal asserts following tx_reset assertion and stays asserted for as long as tx_reset is asserted. This signal deasserts following tx_reset deassertion and remains deasserted for as long as tx_reset is deasserted.
rx_reset	1	Asynchronous	Input	N/A	RX reset input for RX PMAs and RX datapath. Must be kept asserted until rx_reset_ack is asserted. Applies to all RX channels in a F-Tile PMA/FEC Direct PHY Intel FPGA IP instance.
rx_reset_ack	1	Asynchronous	Output	N/A	RX fully in reset indicator. This signal asserts following rx_reset assertion and stays asserted for as long as rx_reset is asserted. This signal deasserts following rx_reset deassertion and remains deasserted for as long as rx_reset is deasserted.

continued...

Name	Width	Domain	Direction	Type	Description
reconfig_pdp_reset	1	Asynchronous	Input	Datapath Avalon Memory Mapped Interface	Reconfiguration Interface Reset
reconfig_xcvr_reset	1	Asynchronous	Input	PMA Avalon Memory Mapped Interface	Active-high synchronous reset. Assert this signal to reset the PMA reconfiguration interface.
tx_ready	1	Asynchronous	Output	N/A	Status port to indicate when TX PMAs and TX datapath are reset successfully and ready for data transfer.
rx_ready	1	Asynchronous	Output	N/A	Status port to indicate when RX PMAs and RX datapath resets are completed, the RX CDRs have locked to data and the recovered line data is ready to be delivered to the parallel interface.
tx_am_gen_start	1	Asynchronous	Output	N/A	When using FEC, indicates when to start sending alignment markers. This clears after tx_am_gen_2x_ack is asserted.
tx_am_gen_2x_ack	1	Asynchronous	Input	N/A	When using FEC, you must indicate to the reset sequencer at least 2 alignment markers were sent since tx_am_gen_start is asserted. This signal should be deasserted after tx_am_gen_start is deasserted.

3.8.5. Status Signals—Descriptions

Table 78. Status Signal Descriptions

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

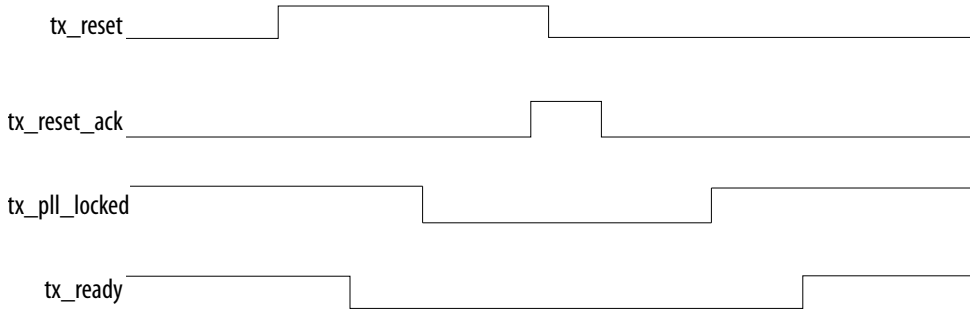
Name	Width	Domain	Direction	Type	Description
tx_pll_locked [*N*-1:0]	*N*	Asynchronous	Output	Direct	TX channel PLL locked signal for both FGT and FHT to reference clock within the PPM threshold status signal for fast/medium or slow PLL. 1'b1 = locked. 1'b0 = not locked.
rx_is_lockedtoref [*N*-1:0]	*N*	Asynchronous	Output	Direct	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.

continued...

Name	Width	Domain	Direction	Type	Description
					Applicable to FGT PMA only. When `lockedtodata` stays high, the `lockedtoref` signal status is insignificant.
rx_is_lockedtodata [*N*-1:0]	*N*	Asynchronous	Output	Direct	RX CDR data lock status signal. • 1'b0: CDR is not locked to data. • 1'b1: CDR is locked to data. Applicable to both FGT and FHT PMA. 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.

3.8.6. Run-time Reset Sequence—TX

Figure 83. Run-time Reset Sequence—TX



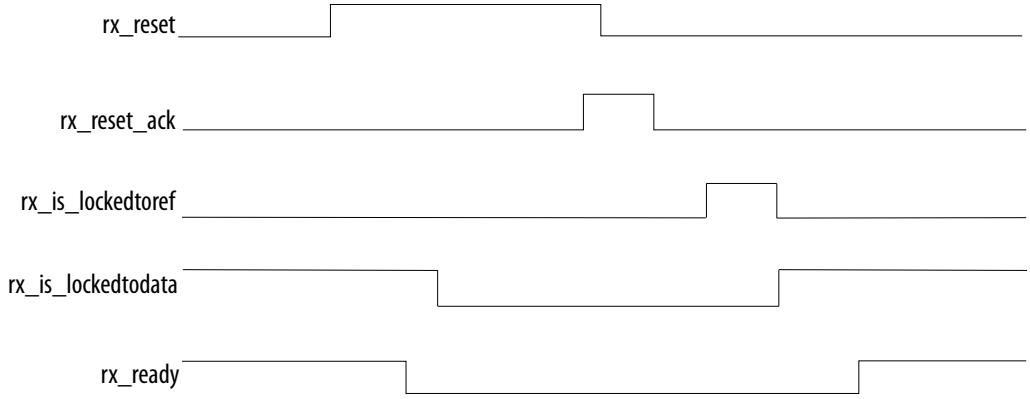
The figure above illustrates the following run-time TX reset sequence:[38]

1. Assert `tx_reset`.
2. `tx_ready` deasserts, indicating that the TX datapath is no longer operational.
3. `tx_pll_locked` deasserts.
4. `tx_reset_ack` asserts, indicating that the TX datapath is fully in reset.
5. You then deassert `tx_reset` to bring TX out of reset.
6. `tx_pll_locked` asserts as the TX PLL locks to the reference clock.
7. `tx_ready` asserts.

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

3.8.7. Run-time Reset Sequence—RX

Figure 84. Run-time Reset Sequence—RX

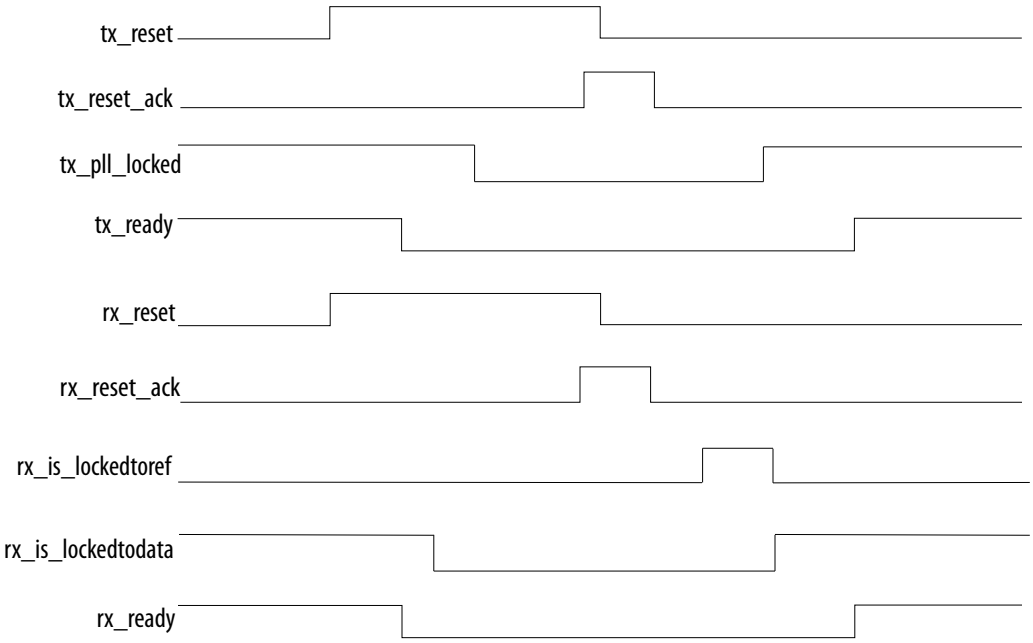


The figure above illustrates the following run-time RX reset sequence:

1. Assert `rx_reset`.
2. `rx_ready` deasserts, indicating that the RX datapath is no longer operational.
3. `rx_is_lockedtodata` deasserts.
4. `rx_reset_ack` asserts, indicating that the RX datapath is fully in reset.
5. You then deassert `rx_reset` to bring RX out of reset.
6. `rx_is_lockedtoref` asserts as the CDR locks to the reference clock.
7. `rx_is_lockedtoref` deasserts and `rx_is_lockedtodata` asserts as the CDR locks to the recovered data.
8. `rx_ready` asserts.

3.8.8. Run-time Reset Sequence—TX + RX

Figure 85. Run-time Reset Sequence—TX + RX



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

1. Assert `tx_reset` and `rx_reset`.
2. `tx_ready` and `rx_ready` deassert, indicating that datapaths are no longer operational.
3. `tx_pll_locked` and `rx_is_lockedtodata` deassert.
4. `tx_reset_ack` and `rx_reset_ack` assert, indicating that the datapaths are fully in reset.
5. You then deassert `tx_reset` and `rx_reset`.
6. `tx_pll_locked` asserts as the PLL locks to the reference clock.
7. `rx_is_lockedtoref` asserts as the CDR locks to the reference clock.
8. `rx_is_lockedtoref` deasserts and `rx_is_lockedtodata` asserts as the CDR locks to the recovered data.
9. `tx_ready` and `rx_ready` assert, indicating that the TX and RX datapaths are ready for use.

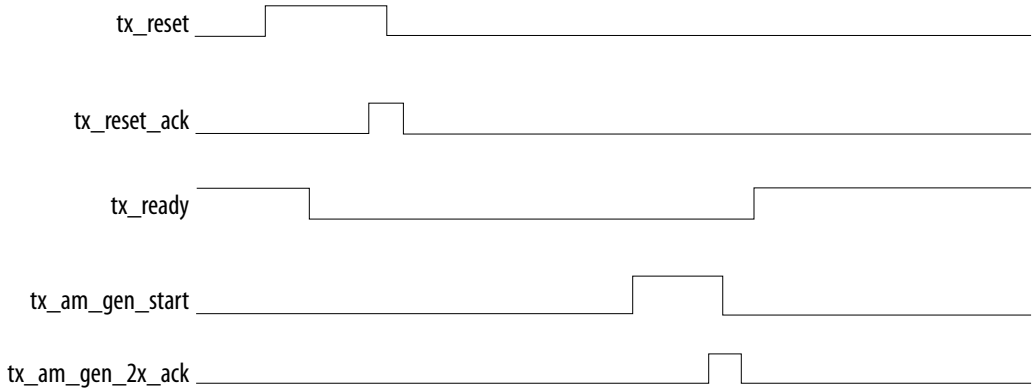
3.8.8.1. Run-time Reset Sequence Approximate Time Durations

The following table provides approximate times for Run-time Reset Sequence—TX + RX. The numbers are provided for general guidance. They are meant to give an idea of the timescale involved for the reset sequence and are subject to change without notice.

Event Sequence	Simulation Max Time	Hardware Max Time
tx_reset_ack deassert to tx_ready assert	740 us	60ms
tx_reset assert to tx_ready deassert	400 us	40ms
tx_reset assert to tx_reset_ack assert	600 us	60ms
rx_reset_ack deassert to rx_ready assert	900 us	70ms
rx_reset assert to rx_ready deassert	400 us	40ms
rx_reset assert to rx_reset_ack assert	1.4 ms	190ms

3.8.9. Run-time Reset Sequence—TX with FEC

Figure 86. 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 `tx_reset`.
2. `tx_ready` deasserts, indicating that the TX datapath is no longer operational.
3. `tx_reset_ack` asserts, indicating that the TX datapath is fully in reset. You then deassert `tx_reset` to bring TX out of reset.
4. `tx_am_gen_start` asserts, you then send at least two alignment markers on the `tx_parallel_data` bus.
5. Assert `tx_am_gen_2x_ack`, indicating that at least two alignment markers have been sent.
6. `tx_am_gen_start` deasserts, you then deassert `tx_am_gen_2x_ack`.
7. `tx_ready` asserts.

3.9. Bonding Implementation

Bonding enables you to minimize skew between channels. You can accomplish bonding between channels by the sharing of TX and RX PMA clocks and synchronizing resets, read enables, and write enables at the PMA interface. The bonding implementation supports up to 16 channels of the FGT PMA, and up to 4 channels of the FHT PMA. In F-tile, there are two components to bonding: PMA bonding and System bonding.

PMA Bonding

PMA bonding is the bonding of streams to and from the PMA channels. For example, on an Ethernet 112G PMA that requires four streams to form a channel, both the TX and RX directions have PMA bonding.

Note: For information on the maximum channel-to-channel skew to expect with TX PMA bonding, refer to the *Intel Agilex 7 Device Data Sheet*.

Note: Use the F-Tile Channel Placement Tool for your PMA channel placement.

System Bonding

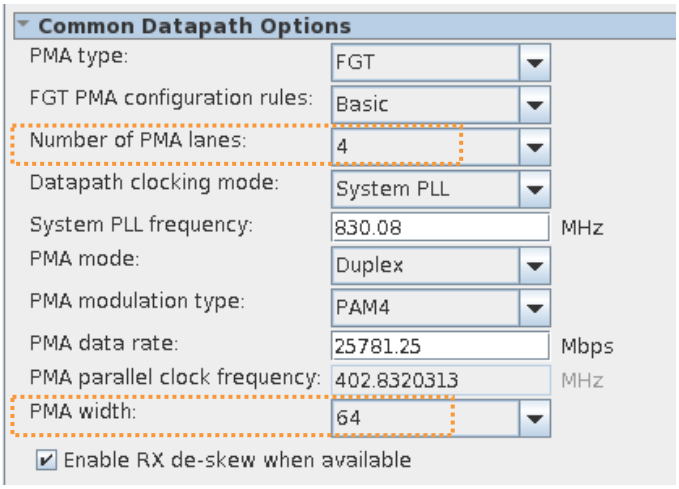
System bonding is the bonding of lanes into a single link. For example, in 400GbE Ethernet, which uses 112G PMAs, each PMA takes four streams. The four streams per PMA are bonded. In turn, these four PMA/stream groupings are all bonded together. Consequently, this 400GbE interface example requires 16 streams. These streams are all bonded together through system bonding.

Note: System bonding in the RX direction is achievable only when using a System PLL as the clock source.

Enabling Bonding in the Parameter Editor

In F-tile, you enable bonding with the **Number of PMA Lanes** and **PMA width** parameters in the **Common Datapath Options** group box in the parameter editor.

Figure 87. Enabling Bonding in the Parameter Editor



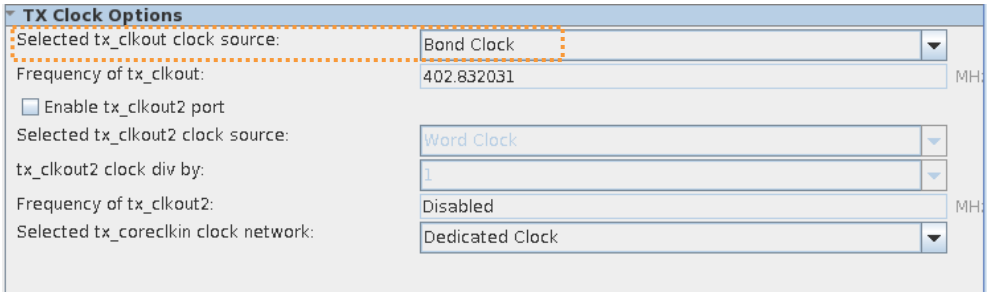
The Intel Quartus Prime software implements bonding automatically, according to your specifications for **Number of PMA Lanes** and **PMA width** parameters. The following table illustrates the parameter settings needed to achieve PMA or system bonding:

Table 79. Parameter Settings to Implement PMA and System Bonding

Bonding Implementation	Number of PMA Lanes	PMA Width
No Bonding	= 1	=< 32-bit
PMA Bonding	= 1	64, 128
System Bonding	> 1	=< 32-bit
System and PMA Bonding	> 1	64, 128

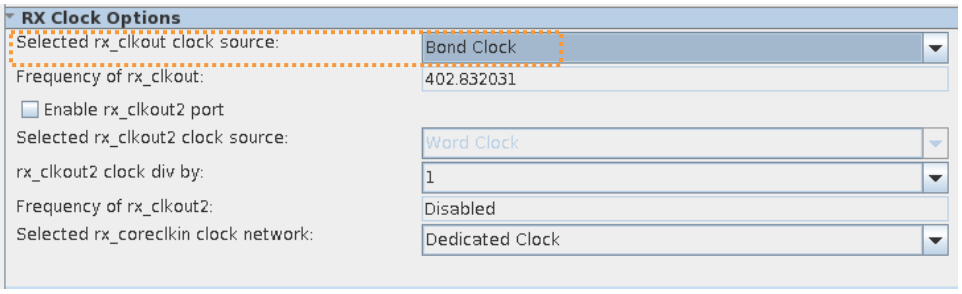
When system bonding is enabled and the **Datapath clocking mode** is set to **PMA**, select **Bond Clock** for the **Selected tx_clkout clock source** parameter in the **TX Clock Options** group box.

Figure 88. TX Clock Options



Similarly, when bonding is enabled and the **Datapath clocking mode** is set to **PMA**, select **Bond Clock** for the **Selected rx_clkout clock source** parameter in the **RX Clock Options** group box.

Figure 89. RX Clock Options



By selecting **Bond Clock** for the **Selected tx_clkout clock source** and **Selected rx_clkout clock source** parameters, you ensure that the primary stream of the bonded group drives the `tx_clkout` and `rx_clkout` sources.

When bonding is enabled and the **Datapath clocking mode** is set to **System PLL**, select either **Sys PLL Clock** or **Sys PLL Clock Div2** for the **Selected tx_clkout clock source** and **Selected rx_clkout clock source** parameters, respectively.

Note: Bonding is not supported across tiles, between FGT and FHT lanes, or between the 200G and 400G hard IPs.

3.10. Independent Port Configurations

There are design scenarios that require completely independent ports with the same IP parameters. To implement this functionality using the current release of the Intel Quartus Prime software, you instantiate multiple copies of the same IP variant in your RTL design.

Later versions of the Intel Quartus Prime software may include the ability to specify the number of independent IP variants in the parameter editor.

3.11. Configuration Registers

You can access the F-Tile PMA registers using the PMA Avalon memory-mapped interface on each lane. You can access the F-Tile PMA/FEC Direct PHY Intel FPGA IP soft CSRs using the datapath Avalon memory-mapped interface.

Write operations to a read-only register field have no effect. Read operations that address a reserved register return an unspecified result. Write operations to reserved registers have an undefined effect. 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. 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 F-tile PMA Register Map contains the reconfiguration register information for:

- PMA and FEC Direct PHY soft CSR registers
- FHT PMA registers
- FGT PMA registers

The following section describes the register map for each area and how to access the registers.

3.11.1. PMA and FEC Direct PHY Soft CSR Register Map

The *PMA and FEC Direct PHY Soft CSR Register Map* allows you to read out the status of the F-Tile 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-reference and lock-to-data status signals. It also allows you to control settings for the PMA hard and soft reset signals.

You must enable the **Enable datapath Avalon interface** and the **Enable Direct PHY soft CSR** settings under the **Datapath Avalon Memory-Mapped Interface** section in the F-Tile PMA/FEC Direct PHY Intel FPGA IP parameter editor to access the soft CSR registers. The datapath Avalon memory-mapped reconfiguration space, starting from offset address 0x800h, contains the F-Tile PMA/FEC Direct PHY Intel FPGA IP soft CSR registers.

Related Information

F-Tile PMA/FEC Direct PHY Intel FPGA IP Register Map

3.11.2. FHT PMA Register Map

The *FHT PMA Register Map* contains the PMA analog registers, user clock settings, debug and loopback registers, PRBS pattern generator and checker registers, error injection and BER measurement registers for the FHT lanes.

You must enable the **Enable PMA Avalon interface** setting under the **PMA Avalon Memory-Mapped Interface** section in the F-Tile PMA/FEC Direct PHY Intel FPGA IP parameter editor to access the FHT PMA registers.

Related Information

F-Tile PMA/FEC Direct PHY Intel FPGA IP Register Map

3.11.3. FGT PMA Register Map

The *FGT PMA Register Map* contains the PMA analog registers, TX PLL counter registers, debug and loopback register information for the FGT lanes.

You must enable the **Enable PMA Avalon interface** setting under the **PMA Avalon Memory-Mapped Interface** section in the F-Tile PMA/FEC Direct PHY Intel FPGA IP parameter editor to access the FGT PMA registers.

Related Information

F-Tile PMA/FEC Direct PHY Intel FPGA IP Register Map

3.11.4. FEC Register Map

The *FEC Register Map* is part of the *F-Tile Ethernet Intel FPGA Hard IP Register Map*.

You must **Enable datapath Avalon interface** setting under the **Datapath Avalon Memory-Mapped Interface** section in the F-Tile PMA/FEC Direct PHY Intel FPGA IP parameter editor to access the FEC registers. In the *F-Tile Ethernet Intel FPGA Hard IP Register Map*, you can filter the Module/Feature column to select FEC to view the FEC registers.

Related Information

- F-Tile Ethernet Intel FPGA Hard IP User Guide
- F-Tile Ethernet Intel FPGA Hard IP Register Map

3.11.5. Lane Offset Address

Lane offset address information is the offset for each lane in the *FHT and FGT PMA Register Maps*.

FHT PMA

The following table shows the FHT PMA lane number to offset address mapping. Word address is `byte address/4`.

Table 80. FHT PMA Lane Number and Offset Address

Lane Number	Lane Base Offset Address (Byte address)
0	0x40000
1	0x48000
2	0x50000
3	0x58000

For example, if you want to control the RX loopback and polarity inversion, refer to the SERDES_LANE_LANE_CTRL_LANE_RX_CTRL register for lane 0 (0x45800) in the register map file and add 0x8000h for each incremental lane, as shown below:

- Lane0 → 0x45800
- Lane1 → 0x4D800
- Lane2 → 0x55800
- Lane3 → 0x5D800

FGT PMA

The following table shows the FGT PMA offset address for each lane within a quad. Word address is `byte address/4`.

Table 81. FGT PMA Lane Number and Offset Address

Lane Number	Lane Base Offset Address (Byte Address)
0	0x40000
1	0x48000
2	0x50000
3	0x58000

For example, you want to update the TX equalizer co-efficients settings for the FGT PMA lanes within a quad, refer to the SRDS_IP_IF_TX1 register for lane 0 (0x47830) in the register map file and add 0x8000h for each incremental lane, as shown below:

- Lane0 → 0x47830
- Lane1 → 0x4F830
- Lane2 → 0x57830
- Lane3 → 0x5F830

Note: The 0x8000h incremental method to read lane 1, lane 2 and lane 3 information is not applicable for the 0xFFFFC register. Refer to *F-Tile PMA/FEC Direct PHY Intel FPGA IP Register Map* for more information.

Note: If your design has more than four FGT PMA lanes that span across multiple FGT quads, refer to Accessing Configuration Registers for more information.

Related Information

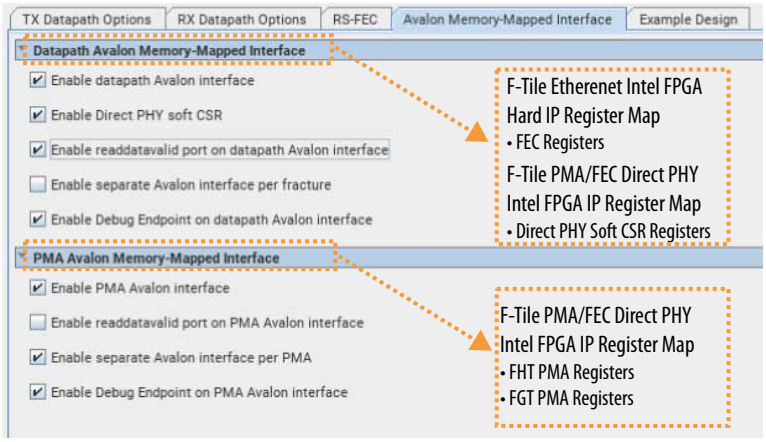
F-Tile PMA/FEC Direct PHY Intel FPGA IP Register Map

3.11.6. Accessing Configuration Registers

This section summarizes how to access the configuration registers listed in the F-Tile 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, FHT PMA registers, and FGT PMA registers. In this section, the terminology *offset address* refers to the address of the configuration registers in the F-Tile PMA/FEC Direct PHY Intel FPGA IP register map.

In order to access all the configuration registers, it is recommended to set the Avalon memory-mapped interface as shown in the following figure.

Figure 90. Recommended Avalon Memory-Mapped Interface Settings



3.11.6.1. Accessing PMA and FEC Direct PHY Soft CSR Registers

For PMA and FEC Direct PHY Soft CSR registers, you can directly use the offset address shown in the F-Tile PMA/FEC Direct PHY Intel FPGA IP register map. For example, to access the PHY reset status register, you must use address 0x80C.

3.11.6.2. Accessing FHT PMA Registers

For FHT PMA registers with offset address less than 0x48000, you must use the following address:

- For the channel on lane 0: offset address
- For the channel on lane 1: offset address + 0x8000
- For the channel on lane 2: offset address + 0x10000
- For the channel on lane 3: offset address + 0x18000

Note: Lane 0, 1, 2, or 3 are the physical locations where the channel is placed at and corresponds to FHT0, FHT1, FHT2, and FHT3 respectively.

For FHT PMA registers with offset address greater than 0x48000 and smaller than 0xFFFFC, you can directly use the offset address provided in the F-Tile PMA/FEC Direct PHY Intel FPGA IP register map.

For FHT PMA registers with offset address 0xFFFFC, you must use the following address:

- For the channel 0: 0xFFFFC
- For the channel 1: 0x1FFFFC
- For the channel 2: 0x2FFFFC
- For the channel 3: 0x3FFFFC

Note: The channel number 0, 1, 2, 3 are the logical number of the PMA lanes. For example, a design with four PMA lanes, has transceiver signals `tx/rx_serial[3:0]`. The signal `tx/rx_serial[0]` is for channel 0, the signal `tx/rx_serial[1]` is for channel 1, the signal `tx/rx_serial[2]` is for channel 2, and the signal `tx/rx_serial[3]` is for channel 3.

3.11.6.3. Accessing FGT PMA Registers

For FGT PMA registers with offset address less than 0x48000, you must use the following address:

- For the channel on lane 0: offset address + channel base address
- For the channel on lane 1: offset address + 0x8000 + channel base address
- For the channel on lane 2: offset address + 0x10000 + channel base address
- For the channel on lane 3: offset address + 0x18000 + channel base address

Note: Lane 0, 1, 2, or 3 are the physical locations where the channel is placed at and corresponds to FGT0, FGT1, FGT2, and FGT3 within each Quad.

The channel base address is as follows:

- For channels 0, 1, 2 or 3: 0x000000
- For channels 4, 5, 6 or 7: 0x400000
- For channels 8, 9, 10 or 11: 0x800000
- For channels 12, 13, 14 or 15: 0xC00000

Note: The channel numbers 0 through 15 are the logical number of the PMA lanes. For example, a design with 16 PMA lanes, has transceiver signals `tx/rx_serial[15:0]`. The signal `tx/rx_serial[0]` is for channel 0, the signal `tx/rx_serial[1]` is for channel 1, the signal `tx/rx_serial[7]` is channel 7 and so on. In addition, the FGT PMA register address depends only on the lane number and channel number and is not related to the Quad number.

FGT PMA Register Access Example 1

This example demonstrates how to access FGT PMA registers of a design with 10 PMA lanes. The placement of the F-Tile PMA/FEC Direct PHY Intel FPGA IP is as follows:

- Channel 0 is placed on Quad 3, Lane 3
- Channel 1 is placed on Quad 3, Lane 2
- Channel 2 is placed on Quad 3, Lane 1
- Channel 3 is placed on Quad 3, Lane 0
- Channel 4 is placed on Quad 2, Lane 3
- Channel 5 is placed on Quad 2, Lane 2

- Channel 6 is placed on Quad 2, Lane 1
- Channel 7 is placed on Quad 2, Lane 0
- Channel 8 is placed on Quad 1, Lane 3
- Channel 9 is placed on Quad 1, Lane 2

To access the TX equalization register with offset address 0x47830, you must use the following address:

- Channel 0: 0x5f830 (0x47830 + 0x18000 + 0x000000)
- Channel 1: 0x57830 (0x47830 + 0x10000 + 0x000000)
- Channel 2: 0x4f830 (0x47830 + 0x8000 + 0x000000)
- Channel 3: 0x47830 (0x47830 + 0x000000)
- Channel 4: 0x45f830 (0x47830 + 0x18000 + 0x400000)
- Channel 5: 0x457830 (0x47830 + 0x10000 + 0x400000)
- Channel 6: 0x44f830 (0x47830 + 0x8000 + 0x400000)
- Channel 7: 0x447830 (0x47830 + 0x400000)
- Channel 8: 0x85f830 (0x47830 + 0x18000 + 0x800000)
- Channel 9: 0x857830 (0x47830 + 0x10000 + 0x800000)

For FGT PMA registers with offset address as 0x9003C or 0x90040, you must use the following address:

- For channels 0, 1, 2 or 3: offset address + 0x000000
- For channels 4, 5, 6 or 7: offset address + 0x400000
- For channels 8, 9, 10 or 11: offset address + 0x800000
- For channels 12, 13, 14 or 15: offset address + 0xC00000

Note: The channel numbers 0 through 15 are the logical number of the PMA lanes. For example, a design with 16 PMA lanes, has transceiver signals `tx/rx_serial[15:0]`. The signal `tx/rx_serial[0]` is for channel 0, the signal `tx/rx_serial[1]` is for channel 1, the signal `tx/rx_serial[7]` is for channel 7 and so on.

For FGT PMA registers with offset address as 0x62000, 0x62004, or 0x62008, you must directly use the offset address provided in the F-Tile PMA/FEC Direct PHY Intel FPGA IP register map.

For FGT PMA registers with offset address greater than 0xF0000 and less than 0xFFFFC, you must directly use the offset address provided in theF-Tile PMA/FEC Direct PHY Intel FPGA IP register map.

For FGT PMA registers with offset address as 0xFFFFC, you must use the following address:

- For channel 0: 0xFFFFC
- For channel 1: 0x1FFFFC
- For channel 2: 0x2FFFFC
- For channel 3: 0x3FFFFC
- For channel 4: 0x4FFFFC
- For channel 5: 0x5FFFFC

- For channel 6: 0x6FFFFC
- For channel 7: 0x7FFFFC
- For channel 8: 0x8FFFFC
- For channel 9: 0x9FFFFC
- For channel 10: 0xAFFFFC
- For channel 11: 0xBFFFFC
- For channel 12: 0xCFFFFC
- For channel 13: 0xDFFFFC
- For channel 14: 0xEFFFFC
- For channel 15: 0xFFFFFC

3.12. Configurable Intel Quartus Prime Software Settings

You can configure the FHT and FGT 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`) to configure the FHT and FGT PMAs:

Note: The settings listed in this section do not apply to the FHT and FGT reference clocks.

3.12.1. FHT PMA Settings

RX invert P and N: To swap RX P and N serial lanes.

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

Example:

```
set_instance_assignment -name HSSI_PARAMETER "rx_invert_p_and_n=RX_INVERT_PN_EN"
-to rx_serial_data[0] -entity top
```

RX termination: This setting is for all lanes.

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

Possible parameter values are:

- RXTERM_OFFSET_P0 (90 Ohms)
- RXTERM_OFFSET_P2 (94.6 Ohms)
- RXTERM_OFFSET_P3 (97.7 Ohms)
- RXTERM_OFFSET_P4 (100 Ohms)
- RXTERM_OFFSET_P5 (102.3 Ohms)
- RXTERM_OFFSET_P6 (105.4 Ohms)
- RXTERM_OFFSET_P7 (107.7 Ohms)

- RXTERM_OFFSET_P8 (110 Ohms)
- RXTERM_OFFSET_M4 (80.8 Ohms)
- RXTERM_OFFSET_M5 (83.1 Ohms)
- RXTERM_OFFSET_M6 (85.4 Ohms)
- RXTERM_OFFSET_M7 (87.7 Ohms)

Example:

```
set_instance_assignment -name HSSI_PARAMETER "rx_termination=RXTERM_OFFSET_P0" -
to rx_serial_data[0] -entity top
```

TX invert P and N: To swap TX P and N serial lanes.

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

Example:

```
set_instance_assignment -name HSSI_PARAMETER
"tx_invert_p_and_n=TX_INVERT_PN_DIS" -to tx_serial_data[0] -entity top
```

TX termination: This setting is for all lanes.

```
set_instance_assignment -name HSSI_PARAMETER
"tx_termination=<TXTERM_OFFSET_NAME>" -to <TX_SERIAL_PIN> -entity
<TOP_LEVEL_NAME>
```

Possible parameter values are:

- TXTERM_OFFSET_M1(96.8 Ohms)
- TXTERM_OFFSET_P0(90 Ohms)

Example:

```
set_instance_assignment -name HSSI_PARAMETER "tx_termination=TXTERM_OFFSET_P0" -
to tx_serial_data[0] -entity top
```

TX out tristate enable: Tx driver tristate enable.

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

Example:

```
set_instance_assignment -name HSSI_PARAMETER
"txout_tristate_en=TXOUT_TRISTATE_EN" -to tx_serial_data[0] -entity top
```

TX equalization:

Post tap: Valid register settings for the parameter value are 0-63

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

Pre tap: Valid register settings for the parameter value are 0-63

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

Main tap: Valid register settings for the parameter value are 0-127

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

Example:

Post tap:

```
set_instance_assignment -name HSSI_PARAMETER "txeq_post_tap_1=0" -to
tx_serial_data[0] -entity top
```

```
set_instance_assignment -name HSSI_PARAMETER "txeq_post_tap_2=0" -to
tx_serial_data[0] -entity top
```

```
set_instance_assignment -name HSSI_PARAMETER "txeq_post_tap_3=0" -to
tx_serial_data[0] -entity top
```

```
set_instance_assignment -name HSSI_PARAMETER "txeq_post_tap_4=0" -to
tx_serial_data[0] -entity top
```

Pre tap:

```
set_instance_assignment -name HSSI_PARAMETER "txeq_pre_tap_3=0" -to
tx_serial_data[0] -entity top
```

```
set_instance_assignment -name HSSI_PARAMETER "txeq_pre_tap_2=0" -to
tx_serial_data[0] -entity top
```

```
set_instance_assignment -name HSSI_PARAMETER "txeq_pre_tap_1=0" -to
tx_serial_data[0] -entity top
```

Main tap:

```
set_instance_assignment -name HSSI_PARAMETER "txeq_main_tap=83" -to
tx_serial_data[0] -entity top
```

3.12.2. FGT PMA Settings

RX on-chip termination:

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

Example:

```
set_instance_assignment -name HSSI_PARAMETER
"rx_onchip_termination=RX_ONCHIP_TERMINATION_R_2" -to rx_serial_data[0] -entity
top
```

Possible parameter values are:

- RX_ONCHIP_TERMINATION_R_1 : 85 Ohms.
- RX_ONCHIP_TERMINATION_R_2 : 100 Ohms.

RX AC coupling:

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

Example:

```
set_instance_assignment -name HSSI_PARAMETER "rx_ac_couple_enable=ENABLE" -to
rx_serial_data[0] -entity top
```

Possible parameter values are:

- ENABLE : when you use on board AC coupling capacitors.
- DISABLE : when you do not use on board AC coupling capacitors.

TX Equalization:

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

Valid parameter values:

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

Examples:

```
set_instance_assignment -name HSSI_PARAMETER "txeq_main_tap=35" -to
tx_serial_data[0] -entity top
```

```
set_instance_assignment -name HSSI_PARAMETER "txeq_pre_tap_1=5" -to
tx_serial_data[0] -entity top
```

```
set_instance_assignment -name HSSI_PARAMETER "txeq_pre_tap_2=0" -to
tx_serial_data[0] -entity top
```

```
set_instance_assignment -name HSSI_PARAMETER "txeq_post_tap_1=0" -to
tx_serial_data[0] -entity top
```

Manual Tunning:

Bypass RX Auto Adaptation

```
set_instance_assignment -name HSSI_PARAMETER "flux_mode=FLUX_MODE_BYPASS" -to
<RX_SERIAL_PIN> -entity <TOP_LEVEL_NAME>
```

```
set_instance_assignment -name HSSI_PARAMETER
"rx_adapt_mode=RX_ADAPT_MODE_STATIC_EQ" -to <RX_SERIAL_PIN> -entity
<TOP_LEVEL_NAME>
```

RX Manual Equalization

- VGA:

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

  Valid parameter values are 0-63.
- High-frequency boost:

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

  Valid parameter values are 0-63.
- DFE Tap 1:

  ```
  set_instance_assignment -name HSSI_PARAMETER
  "rxeq_dfe_data_tap_1=<parameter_value>" -to  <RX_SERIAL_PIN> -entity
  <TOP_LEVEL_NAME>
  ```

  Valid parameter values are 0-63.

VSR is set when insertion loss < 10dB, RX auto adaptation is ON for >= 23 Gbps NRZ or any PAM4 rate:

- For Insertion Loss <8dB:

```
set_instance_assignment -name HSSI_PARAMETER "vsr_mode=VSR_MODE_LOW_LOSS" -
to <RX_SERIAL_PIN> -entity < TOP_LEVEL_NAME>
```

- For Insertion Loss > 8dB but < 10dB:

```
set_instance_assignment -name HSSI_PARAMETER "vsr_mode=VSR_MODE_HIGH_LOSS" -
to <RX_SERIAL_PIN> -entity < TOP_LEVEL_NAME>
```

- Else VSR_MODE_DISABLE is set by default by the Intel Quartus Prime Pro Edition software for rates below < 23 Gbps where VSR is not required.

3.13. Configuring the F-Tile PMA/FEC Direct PHY Intel FPGA IP for Hardware Testing

This section details the steps you should follow to configure the F-Tile PMA/FEC Direct PHY Intel FPGA IP in order to bring-up the FHT or FGT PMA for hardware testing using System Console in the Intel Quartus Prime software. You can configure the 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 the Debug Endpoint Avalon interface within the F-Tile PMA/FEC Direct PHY Intel FPGA IP.
- Using JTAG to Avalon Master Bridge Intel FPGA IP instantiated from the IP Catalog.

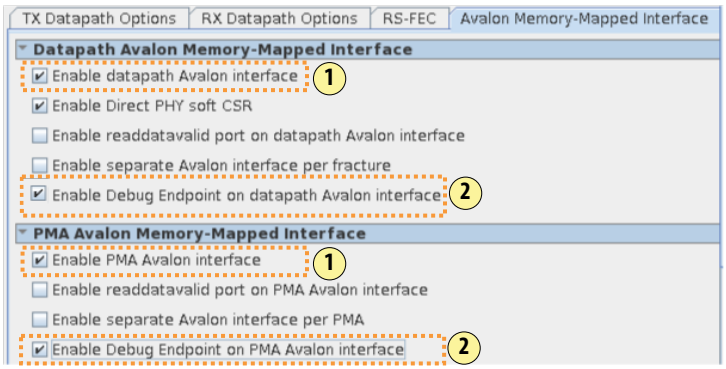
3.13.1. Using Debug Endpoint Interface within the F-Tile PMA/FEC Direct PHY Intel FPGA IP

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

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

1. Enable the **Enable datapath Avalon interface** and **Enable PMA Avalon interface** options in the **Avalon Memory-Mapped Interface** tab of the F-Tile PMA/FEC Direct PHY Intel FPGA IP parameter editor.
2. Enable the **Enable Debug Endpoint on datapath Avalon interface** option and **Enable Debug Endpoint on PMA Avalon interface** option in the **Avalon Memory-Mapped Interface** tab of the F-Tile PMA/FEC Direct PHY Intel FPGA IP parameter editor.

Figure 91. IP Parameter Editor



3. Connect the clock and reset signals to the `reconfig_pdp_clk` and `reconfig_pdp_reset` ports of the datapath reconfiguration interface.
4. Connect the other datapath reconfiguration interface signals:
   - `reconfig_pdp_write`
   - `reconfig_pdp_read`
   - `reconfig_pdp_address`
   - `reconfig_pdp_writedata`
   - `reconfig_pdp_readdata`
   - `reconfig_pdp_byteenable`
   - `reconfig_pdp_readdatavalid`
   - `reconfig_pdp_waitrequest`

   to ground, assuming no FPGA core logic controls the reconfiguration interface.
5. Follow the same connection guidelines in steps 3 and 4 for the `reconfig_xcvr*` PMA interface signals.

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

3.13.1.1. RTL Connection Example for Debug Endpoint Avalon Interface

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

Example datapath reconfiguration interface connections for a 16 PMA lanes design:

```
.reconfig_pdp_clk           ( 100MHz         ),
.reconfig_pdp_reset         ( reconfig_reset ),
.reconfig_pdp_write         ( 1'b0           ),
.reconfig_pdp_read          ( 1'b0           ),
.reconfig_pdp_address       ( 18'b0          ),
.reconfig_pdp_byteenable    ( 4'b0           ),
.reconfig_pdp_writedata     ( 32'b0          ),
.reconfig_pdp_readdata      ( ),
.reconfig_pdp_waitrequest   ( )
```

Example PMA reconfiguration interface connections for a 16 PMA lanes design:

```
.reconfig_xcvr_clk          ( 100MHz          ),
.reconfig_xcvr_reset        ( reconfig_reset  ),
.reconfig_xcvr_write        ( 1'b0            ),
.reconfig_xcvr_read         ( 1'b0            ),
.reconfig_xcvr_address      ( 22'b0           ),
.reconfig_xcvr_byteenable   ( 4'b0            ),
.reconfig_xcvr_writedata    ( 32'b0           ),
.reconfig_xcvr_readdata     ( ),
.reconfig_xcvr_waitrequest  ( )
```

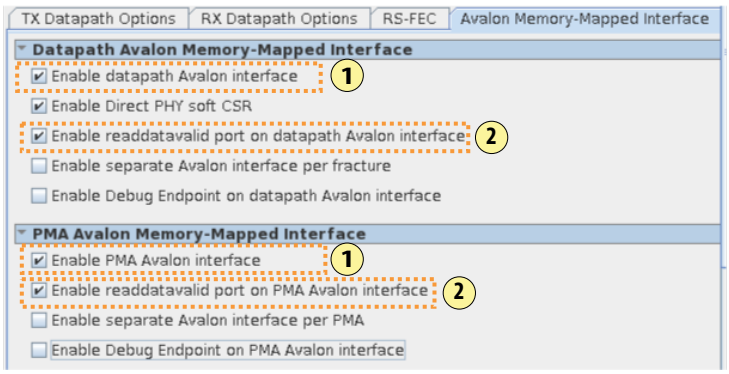
3.13.2. Using JTAG to Avalon Master Bridge Intel FPGA IP

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

To Enable the JTAG to Avalon Master Bridge Intel FPGA IP Interface

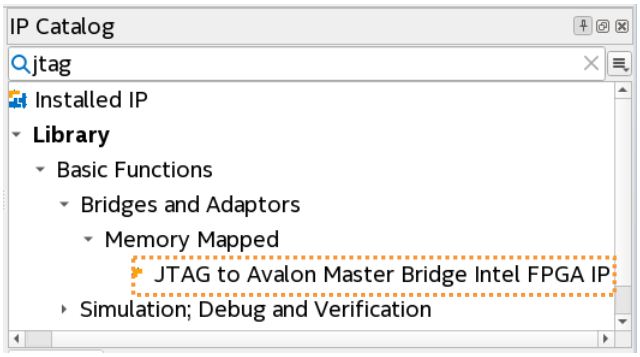
1. Enable the **Enable datapath Avalon interface** and **Enable PMA Avalon interface** options in the **Avalon Memory-Mapped Interface** tab of the F-Tile PMA/FEC Direct PHY Intel FPGA IP parameter editor.
2. Enable the **Enable readdatavalid port on datapath Avalon interface** and **Enable readdatavalid port on PMA Avalon interface** options in the **Avalon Memory-Mapped Interface** tab of the F-Tile PMA/FEC Direct PHY Intel FPGA IP parameter editor.

Figure 92. IP Parameter Editor



3. Instantiate two JTAG to Avalon Master Bridge Intel FPGA IP instances from the IP Catalog. The first instance to interface with datapath Avalon interface, and the second instance to interface with the PMA Avalon interface.

Figure 93. IP Catalog



4. Connect the clock and reset signals to the `reconfig_pdp_clk` and `reconfig_pdp_reset` ports of the datapath reconfiguration interface.
5. Connect the other datapath reconfiguration interface signals:
   - `reconfig_pdp_write`
   - `reconfig_pdp_read`
   - `reconfig_pdp_address`
   - `reconfig_pdp_writedata`
   - `reconfig_pdp_readdata`
   - `reconfig_pdp_byteenable`
   - `reconfig_pdp_readdatavalid`
   - `reconfig_pdp_waitrequest`

   To the equivalent JTAG to Avalon Master Bridge Intel FPGA IP reconfiguration signals.
6. Follow the same connection guidelines as in steps 4 and 5 for the `reconfig_xcvr*` PMA interface signals.

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

The following examples show the RTL connections for a two lane FGT PMA channel implementation.

The `reconfig_pdb_address` and `reconfig_xcvr_address` address bus width is:

```
[14:0] reconfig_pdp_address,
[18:0] reconfig_xcvr_address,
```

You set the address parameters for the datapath and PMA interface in your design file as follows:

```
parameter  pdp_addr_width = 15,
parameter  xcvr_addr_width = 19,
```

Note: The F-Tile PMA/FEC Direct PHY Intel FPGA IP uses word addressing format for both the datapath and PMA 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 bus by two bits, as shown in the following examples.

Example datapath reconfiguration interface connections:

```
.reconfig_pdp_clk           ( 100MHz                    ),
.reconfig_pdp_reset         ( reconfig_reset            ),
.reconfig_pdp_write         ( pdb_write_bridge          ),
.reconfig_pdp_read          ( pdb_read_bridge           ),
.reconfig_pdp_address       ( pdb_address_bridge [pdp_addr_width + 1: 2]),
.reconfig_pdp_byteenable    ( pdb_byteenable_bridge     ),
.reconfig_pdp_writedata     ( pdb_writedata_bridge      ),
.reconfig_pdp_readdata      ( pdb_readdata_bridge       ),
.reconfig_pdp_readdatavalid ( pdb_readdatavalid_bridge ),
.reconfig_pdp_waitrequest   ( pdb_waitrequest_bridge    )
```

Example PMA reconfiguration interface connections:

```
.reconfig_xcvr_clk           ( 100MHz                     ),
.reconfig_xcvr_reset         ( reconfig_reset             ),
.reconfig_xcvr_write         ( xcvr_write_bridge          ),
.reconfig_xcvr_read          ( xcvr_write_bridge          ),
.reconfig_xcvr_address       ( xcvr_address_bridge [xcvr_addr_width + 1: 2]),
.reconfig_xcvr_byteenable    ( xcvr_byteenable_bridge     ),
.reconfig_xcvr_writedata     ( xcvr_writedata_bridge      ),
.reconfig_xcvr_readdata      ( xcvr_readdata_bridge       ),
.reconfig_xcvr_readdatavalid ( xcvr_readdatavalid_bridge ),
.reconfig_xcvr_waitrequest   ( xcvr_waitrequest_bridge    )
```

3.14. Hardware Configuration Using the Avalon Memory-Mapped Interface

You can configure the FHT and FGT PMAs in various modes using the Avalon memory-mapped interface.

3.14.1. FHT PMA

The following section describes hardware configuration for the FHT PMA using the Avalon memory-mapped interface.

3.14.1.1. Enabling Loopback

Use the following register settings to enable FHT PMA loopback modes.

Table 82. Loopback Mode Register Settings

Loopback Mode	Register	Description
Serial internal loopback	0x45800[14]	Set `1'b1` to enable serial internal loopback. Set `1'b0` to disable serial internal loopback.
Parallel TX to RX loopback	0x45800[0]	Set `1'b1` to enable parallel TX to RX loopback. Set `1'b0` to disable parallel TX to RX loopback.
Reverse parallel RX to TX loopback	0x45804[19]	Set `1'b1` to enable reverse parallel RX to TX loopback. Set `1'b0` to disable reverse parallel RX to TX loopback.

3.14.1.2. Enabling the PRBS Generator and Verifier

The PRBS Generator and Verifier provides a method to debug and validate your PMA links.

To enable the PRBS Generator and Verifier, follow these steps:

1. Set `car_tx_clk_src_sel` (0x60000[2]) to `1'b1`.
2. Set `cfg_tx_bus_take_dft` (0x45804[0]) to `1'b1`. If using multi-lanes, set `1'b1` to all lanes.
3. Set `cfg_lane_tx_prbs_en` (0x42934[0]) to `1'b1`. If using multi-lanes, set `1'b1` to all lanes.
4. Specify the PRBS generator pattern `cfg_lane_tx_prbs_mode` (0x42934[4:1]). If using multi-lanes, specify for all lanes.
5. Set `cfg_lane_tx_prbs_init` (0x4293C[0]) to `1'b1`. If using multi-lanes, set `1'b1` to all lanes.
6. Set `cfg_dft_rx_prbs_common_en` (0x42930[0]) to `1'b1`. If using multi-lanes, set `1'b1` to all lanes.
7. Specify the PRBS verifier pattern, `cfg_dft_rx_prbs_sel` (0x42930[4:1]). If using multi-lanes, specify for all lanes.
8. Set `cfg_rx_dft_data_sel` (0x42930[6:5]) to `2'b00`. If using multi-lanes, set `2'b00` to all lanes.
9. Set `cfg_ber_symb_cnt_limit_lsb` (0x428EC[31:0]). If using multi-lanes, set for all lanes.
10. Set `cfg_ber_symb_cnt_limit_msb` (0x428F0[31:0]). If using multi-lanes, set for all lanes.
11. Set `cfg_dft_ber_count_en` (0x428DC[0]) to `1'b1`. If using multi-lanes, set `1'b1` to all lanes.
12. Set cfg_dft_ber_count_mode (0x428DC[2:1]) to `2'b10`. If using multi-lanes, set `2'b10` to all lanes.

3.14.1.3. TX Equalizer Settings

The TX equalizer settings provides a way to tune your PMA TX buffer to optimize link performance.

To update the TX equalizer settings, follow these steps:

1. Set `csr_txffe_coeff_load` (0x45080[0]) to `1'b0`.
2. Set TX equalizer co-efficients to valid settings:
   - TX equalizer pre-cursor 3 register `csr_txffe_coeff_p5`(0x45084[23:18]).
   - TX equalizer pre-cursor 2 register `csr_txffe_coeff_m2`(0x45080[7:2]).
   - TX equalizer pre-cursor 1 register `csr_txffe_coeff_m1`(0x45080[13:8]).
   - TX equalizer main cursor register `csr_txffe_coeff_0`(0x45080[20:14]).
   - TX equalizer post-cursor 1 register `csr_txffe_coeff_p1`(0x45080[26:21]).
   - TX equalizer post-cursor 2 register `csr_txffe_coeff_p2`(0x45084[5:0]).
   - TX equalizer post-cursor 3 register `csr_txffe_coeff_p3`(0x45084[11:6])
   - TX equalizer post-cursor 4 register `csr_txffe_coeff_p4`(0x45084[17:12]).
3. Toggle `csr_txffe_coeff_load` (0x45080[0]) to `1'b1` and back to `1'b0`.

Note: For the range of TX PMA equalizer parameters, refer to FHT Transmitter PMA Equalizer Parameters for NRZ and PAM4 Modes.

Table 83. Main Cursor (C0) Real Co-efficient Values

Main Cursor (C0) – Register 0x45080[20:14] Settings (decimal)	Real Co-efficient Values
0	0
1	0.5
2	1
…	…
82	41
83	41.5

Table 84. Pre-Cursor (C-1) and Post-Cursor (C1) Real Co-efficient Values

Pre-Cursor (C-1) – Register 0x45080[13:8] Post-Cursor (C1) – Register 0x45080[26:21] Settings (decimal)	Real Co-efficient Values
0	0
1	0.5
2	1
…	…
30	15
31	15.5
32	-16
33	-15.5
…	…
62	-1
63	-0.5

Table 85. Pre-Cursor (C-2, C-3) and Post-Cursor (C2, C3, C4) Real Co-efficient Values

Pre-Cursor (C-2) – Register 0x45080 [7:2] Pre-Cursor (C-3) – Register 0x45084[23:18] Post-Cursor (C2) – Register 0x45084[5:0] Post-Cursor (C3) – Register 0x45084[11:6] Post-Cursor (C4) – Register 0x45084[17:12] Settings (decimal)	Real Co-efficient Values
0	0
1	0.25
2	0.5
…	…
30	7.5
31	7.75

continued...

Pre-Cursor (C-2) – Register 0x45080 [7:2] Pre-Cursor (C-3) – Register 0x45084[23:18] Post-Cursor (C2) – Register 0x45084[5:0] Post-Cursor (C3) – Register 0x45084[11:6] Post-Cursor (C4) – Register 0x45084[17:12] Settings (decimal)	Real Co-efficient Values
32	-8
33	-7.75
…	…
62	-0.5
63	-0.25

3.14.1.4. TX Error Injection

FHT PMA supports programmable number of error injection into the TX datapath.

To configure the TX datapath error injection, follow these steps:

1. Write `cfg_tx_err_inj_mask_cfg` (0x45808[21:6]) to a value that you want to mask the TX data with and cause an error injection.
2. Write `cfg_tx_err_inj_mask_load` (0x45810[0]) to `1'b1`. (self-clears to `1'b0`).

   Note: When you write a `1'b1` to this bit, the internal 128-bit error mask register shifts left by 16 bits, and the new 16-bit mask value replaces the LSB bits in the register. The 64-bit and 32-bit width modes use only the MSBs of the datapath. Therefore, you must load the mask multiple times to shift it into the MSB bits.
3. Write `err inj block-write cfg_tx_err_inj_en` (0x45808[0]) to `1'b1` to enable error injection.
4. Write `cfg_tx_err_inj_trig` (0x4580C[0]) to `1'b1` (self-clears to `1'b0`) to inject the errors. Every time you write a `1'b1` to this bit, the 128 bit of datapath is XORed with 128 bits of error mask register.
5. Repeat steps 1 and 2 to inject more errors.

3.14.1.5. RX Reconvergence

RX reconvergence is performed after enabling internal loopback, after cable removal or after changing TX equalizer settings.

Write `1'b1` to `rx_reconverge` (0x4585C[14]) to enable RX reconvergence.

3.14.1.6. Preserving Unused Lanes

FHT PMA supports preservation of unused PMA lanes that you plan to use in the next iteration of your FPGA design.

Write `4'b1111` to `cfg_preserve_enable` (0xF0030[3:0]) to preserve all the unused lanes in a FHT PMA.

Note: Set this register to zeros for normal operation. This is a common register for all four lanes. LSB is for lane 0 and MSB is for lane 3.

3.14.2. FGT PMA

You can perform hardware configuration using Avalon memory-mapped interface for FGT PMAs using these methods:

- Direct Register Method
- FGT Attribute Access Method

3.14.2.1. Direct Register Method

Using the direct register method, you update the FGT PMA registers directly to configure the FGT PMA.

Note: In this section, the offset addresses are shown for Lane 0. To get addresses for your appropriate channel, refer to the section: Accessing FGT PMA Registers.

For example, you can configure the TX equalizer co-efficient values using the direct register method.

3.14.2.1.1. Direct Register Method Examples

The following examples demonstrate the direct register method to configure the FGT PMA.

TX Equalizer Co-efficients

To set the TX equalizer co-efficients:

- Write the TX equalizer pre_tap_2 register (0x47830[18:16]) with valid value.
- Write the TX equalizer pre_tap_1 register (0x47830[9:5]) with valid value.
- Write the TX equalizer main_tap register (0x47830[15:10]) with valid value.
- Write the TX equalizer post_tap_1 register (0x47830[4:0]) with valid value.

Mute TX Output

To mute TX output (make TX output 0v):

- Set 0x41750[25:24] to 2'b11

To unmute TX output:

- Set 0x41750[25:24] to 2'b00

Internal Serial Loopback

To enable internal serial loopback[39]:

- Set 0x41418[31] to 0x0
- Set 0x41420[25] to 0x1
- Set 0x41418[29] to 0x1
- Set 0x41418[31] to 0x1

To disable internal serial loopback:

(39) The sequence is valid only when RX manual tuning is used (RX auto adaptation is bypassed). If RX auto adaptation is used, use the FGT attribute access method.

- Set 0x41418[31] to 0x0
- Set 0x41418[29] to 0x0
- Set 0x41420[25] t0 0x0

Reverse Parallel Loopback

To enable the Reverse Parallel Loopback:

- Write 0x1 to 0x41414[29]
- Write 0x1 to 0x4141C[30]
- Write 0x1 to 0x41418[31]

To disable the Reverse Parallel Loopback:

- Write 0x0 to 0x41414[29]
- Write 0x0 to 0x4141C[30]
- Write 0x0 to 0x41418[31]

TX to RX Parallel Loopback

To enable the TX to RX Parallel Loopback:

- Write 0x1 to 0x416A4[8]
- Write 0x1 to 0x41418[31]

To disable the TX to RX Parallel Loopback:

- Write 0x0 to 0x416A4[8]
- Write 0x0 to 0x41418[31]

Polarity Inversion

TX polarity inversion[39]:

- Write 0x1 to 0x41428[7]

TX polarity inversion revert back:

- Write 0x0 to 0x41428[7]

RX polarity inversion:

- Write 0x1 to 0x41428[6]

RX polarity inversion revert back:

- Write 0x0 to 0x41428[6]

Measuring the Bit Error Rate (BER) with FGT PMAs

1. Check that the RX link is ready for the desired lane:
   a. Read 0x814[31:16] to confirm that the corresponding lane's `rx_cdr_locked2data` = 1
2. Assign the PRBS pattern value:
   a. For TX:
      i. Set valid values to 0x416AC[31:28]
   b. For RX:

i. Set valid values to 0x41428[3:0]

c. Valid values for PRBS pattern:
- UDP : 0x0
- PRBS7 : 0x1
- PRBS9 : 0x2
- PRBS11 : 0x3
- PRBS13 : 0x4
- PRBS15 : 0x5
- PRBS23 : 0x6
- PRBS28 : 0x7
- PRBS31 : 0x8
- QPRBS13 : 0x9
- PRBS13Q : 0xa
- PRBS31Q : 0xb
- SSPR : 0xc
- SSPR1 : 0xd
- SSPRQ : 0xe

3. BER Start:
   a. Write 0x1 to 0x416AC[23]
   b. Write 0x1 to 0x41424[26]
   c. Write 0x3 to 0x4176C[28:27]
   d. Write 0x3 to 0x415B4[19:18]
4. BER Count:
   a. Read from 0x41444[31:0]
5. BER Stop:
   a. Write 0x0 to 0x416AC[23]
   b. Write 0x0 to 0x41424[26]
   c. Write 0x0 to 0x4176C[28:27]
   d. Write 0x0 to 0x415B4[19:18]
6. To check overflow, read 0x4143C[21]
7. To clear the counter, toggle 0x415B4[19:18]:
   a. Write 0x3 to 0x415B4[19:18]
   b. Write 0x0 to 0x415B4[19:18]

3.14.2.2. FGT Attribute Access Method

Using the FGT attribute access method, you update the FGT PMA registers to configure hardware with a specific sequence of commands.

For example, you can configure serial internal loopback, PRBS generator and verifier using the FGT attribute access method. The FGT attribute access method consists of 4 steps in a sequence as shown below:

1. Write data value to LINK_MNG_SIDE_CPI_REGS register to assert a service request.
2. Read PHY_SIDE_CPI_REGS register to confirm the request has been acknowledged and completed; if not, repeat this step.
3. Write data value to LINK_MNG_SIDE_CPI_REGS register to deassert the service request.
4. Read PHY_SIDE_CPI_REGS register to confirm the request in step 3 has been acknowledged; if not, repeat this step.

Table 86. FGT Attribute Access Addresses for JTAG Master that Controls 16 channels

Channels	LINK_MNG_SIDE_CPI_REGS Address	PHY_SIDE_CPI_REGS Address
Channel 0 or 1 or 2 or 3	0x0009003c	0x00090040
Channel 4 or 5 or 6 or 7	0x0049003c	0x00490040
Channel 8 or 9 or 10 or 11	0x0089003c	0x00890040
Channel 12 or 13 or 14 or 15	0x00C9003c	0x00C90040

Table 87. FGT Attribute Access Data Value 1

	Serial Loopback	TX and RX PRBS Selection	Polarity Setup	BER Measurement	Start/Stop Test
Data field[31:16]	Enable: 0x6 Disable: 0x0	PRBS7: 0x208 PRBS9: 0x249 PRBS11: 0x28A PRBS13: 0x965 PRBS23: 0x2CB PRBS31: 0x30C QPRBS13: 0x34D PRBS13Q: 0x820 PRBS31Q: 0x861 SSPR: 0x8A2 SSPR1: 0x8E3 SSPRQ: 0x924	Reverse: 0x1 Revert back: 0x0	0x14	Start: 0x20 Stop: 0x21
Option field [15:12]	Bit [15] SERVICE_REQ to indicate a request: 0 = no request, 1 = service requested. Bit [14] RESET: 0 = not in reset, 1 = in reset. Bit [13] SET_GET: 0 = GET parameters, 1 = SET parameters. Bit [12]: reserved				
Lane number field[11:8]	Use 0xFFFFC[1:0], 0x1FFFFC[1:0]… 0xFFFFFC[1:0] to read back logical lane 0, 1 until lane 15's physical lane number. • If return value is 2'b00, physical lane is 0 • If return value is 2'b01, physical lane is 1 • If return value is 2'b10, physical lane is 2 • If return value is 2'b11, physical lane is 3				
Opcode field[7:0]	0x40	0x41	TX polarity: 0x65 RX polarity: 0x66	0x45	0x0F *Note:* 0x0F is not equivalent to 0xF

Table 88. FGT Attribute Access Data Value 2

	Get Status	Error Number to Inject	Enable Error Injection	Read Results	Finish BER Measurement
Data field[31:16]	0x0	0x[Error_Num]	0x23	0x0	0x0
Option field [15:12]	Bit [15] SERVICE_REQ to indicate a request: 0 = no request, 1 = service requested. Bit [14] RESET: 0 = not in reset, 1 = in reset. Bit [13] SET_GET: 0 = GET parameters, 1 = SET parameters. Bit [12]: reserved				
Lane number field[11:8]	Use 0xFFFFC[1:0], 0x1FFFFC[1:0]… 0xFFFFFC[1:0] to read back logical lane 0, 1 until lane 15's physical lane number. • If return value is 2'b00, physical lane is 0 • If return value is 2'b01, physical lane is 1 • If return value is 2'b10, physical lane is 2 • If return value is 2'b11, physical lane is 3				
Opcode field[7:0]	0x49: Get Test status 0x0D: Get PMA status	0x42	0x0F *Note:* 0x0F is not equivalent to 0xF	• LSB: 0x4A • Middle: 0x4B • MSB: 0x4C	0x41

You can create a function to write data, or read to and from FGT attribute access addresses. The data is comprised of `data field[31:16]`, `option field[15:12]`, `lane number field[11:8]`, and `opcode field[7:0]`. The following examples use the `tcl` process as shown below:

```
proc attribute_access {{data field} {option field} {lane number field} {opcode field}}
```

You can use any programming language to perform the read and writes. For the other FGT PMA lanes, refer to FGT Attribute Access Addresses for JTAG Master that Controls 16 channels for `LINK_MNG_SIDE_CPI_REGS` and `PHY_SIDE_CPI_REGS`, and refer to FGT Attribute Access Data Value 1 for lane number field information.

3.14.2.2.1. FGT Attribute Access Method Example 1

The following example describes the steps to enable and disable serial loopback.

1. To enable serial loopback:
   a. Assert RX reset.
   b. Write `0x6A040` to address 0x9003C.

      `6`: data field for serial loopback enable.

      `A`: option field to request service, not set in reset and set parameters.

      `0`: lane number field for physical lane 0.

      `40`: opcode field for serial loopback.
   c. Poll address 0x90040 until bit `14 = 0` and bit `15 = 1`.
   d. Write `0x62040` to address 0x9003C.

      `6`: data field for serial loopback enable.

      `2`: option field to de-assert service request, not set in reset and set parameters.

      `0`: lane number field for physical lane 0.

40: opcode field for serial loopback.

e. Poll address 0x90040 until bit 14 = 0 and bit 15 = 0.

f. Deassert RX reset.

2. To disable serial loopback:

a. Assert RX reset.

b. Write 0x0A040 to address 0x9003C.

0: data field for serial loopback disable.

A: option field to request service, not set in reset and set parameters.

0: lane number field for physical lane 0.

40: opcode field for serial loopback.

c. Poll address 0x90040 until bit 14 = 0 and bit 15 = 1.

d. Write 0x02040 to address 0x9003C.

0: data field for serial loopback disable.

2: option field to de-assert service request, not set in reset and set parameters.

0: lane number field for physical lane 0.

40: opcode field for serial loopback.

e. Poll address 0x90040 until bit 14 = 0 and bit 15 = 0.

f. Deassert RX reset.

3.14.2.2.2. FGT Attribute Access Method Example 2

The following example demonstrates the steps to enable the FGT PMA PRBS checker and generator for logical lane 0, when you configure the FGT PMA in internal serial loopback mode in physical lane 0 of a quad, using the FGT attribute access method.

1. Assert rx_reset.
2. Enable serial loopback:
   a. Write 0x6A040 to address 0x9003C.
   b. Poll address 0x90040 until bit 14 = 0 and bit 15 = 1.
   c. Write 0x62040 to address 0x9003C.
   d. Poll address 0x90040 until bit 14 = 0 and bit 15 = 0.
3. Deassert rx_reset.
4. Confirm the channel is in serial loopback:
   a. Poll register 0x4781C; bit 1 should be high if serial loopback is enabled.
5. Check the FGT PMA's status:
   a. Write 0x800D to address 0x9003C.
   b. Poll address 0x90040 until bit 15 = 1; bit 16 should also be high if the channel is located in physical local 0.

Note: bit 16, `rx_ready` is for physical local lane 0
bit 17, `rx_ready` is for physical local lane 1
bit 18, `rx_ready` is for physical local lane 2
bit 19, `rx_ready` is for physical local lane 3

c. Write `0x000D` to address 0x9003C.
d. Poll address 0x90040 until bit `15 = 0`.

6. Set the PRBS31 pattern for both the TX and RX PMAs:
   a. Write `0x30CA041` to address 0x9003C.
   b. Poll address 0x90040 until bit `15 = 1`.
   c. Write `0x30C2041` to address 0x9003C.
   d. Poll address 0x90040 until bit `15 = 0`.
7. Set up the PMA to count the number of bit errors:
   a. Write `0x14A045` to address 0x9003C.
   b. Poll address 0x90040 until bit `15 = 1`.
   c. Write `0x142045` to address 0x9003C.
   d. Poll address 0x90040 until bit `15 = 0`.
8. Start the test:
   a. Write `0x20A00F` to address 0x9003C.
   b. Poll address 0x90040 until bit `15 = 1`.
   c. Write `0x20200F` to address 0x9003C.
   d. Poll address 0x90040 until bit `15 = 0`.
9. Check that the test is running:
   a. Write `0x8049` to address 0x9003C.
   b. Poll address 0x90040 until bit `15 = 1`; bits `25:24` should be `0x1` to indicate the test is running.[(40)]
   c. Write `0x0049` to address 0x9003C.
   d. Poll address 0x90040 until bit `15 = 0`.
10. Set up the PRBS generator to inject errors:

(40) *Note:* 0x90040[25:24] status values:

- `0x0`: Idle
- `0x1`: Test running
- `0x2`: Test stopped-execution failure
- `0x3`: Test stopped-execution completed successfully

a. Write 0x123A042 to address 0x9003C to inject 0x123 errors.
b. Poll address 0x90040 until bit 15 = 1.
c. Write 0x1232042 to address 0x9003C.
d. Poll address 0x90040 until bit 15 = 0.

11. Tell the PRBS generator to inject errors:
    a. Write 0x23A00F to address 0x9003C.
    b. Poll address 0x90040 until bit 15 = 1.
    c. Write 0x23200F to address 0x9003C.
    d. Poll address 0x90040 until bit 15 = 0.
12. Stop the BER test:
    a. Write 0x21A00F to address 0x9003C.
    b. Poll address 0x90040 until bit 15 = 1.
    c. Write 0x21200F to address 0x9003C.
    d. Poll address 0x90040 until bit 15 = 0.
13. Check the test completed successfully:
    a. Write 0x8049 to address 0x9003C.
    b. Poll address 0x90040 until bit 15 = 1; bits 25:24 should be 0x3.(40)
    c. Write 0x0049 to address 0x9003C.
    d. Poll address 0x90040 until bit 15 = 0.
14. Read out the 12 LSB of the error count:
    a. Write 0x804A to address 0x9003C.
    b. Poll address 0x90040 until bit 15 = 1; bits 27:16 represent the 12 LSBs of the error count.
    c. Write 0x004A to address 0x9003C.
    d. Poll address 0x90040 until bit 15 = 0.
15. Read out bits 27:12 of the error count:
    a. Write 0x804B to address 0x9003C.
    b. Poll address 0x90040 until bit 15 = 1; bits 31:16 represent bits 27:12 of the error count.
    c. Write 0x004B to address 0x9003C.
    d. Poll address 0x90040 until bit 15 = 0.
16. Read out bits 31:28 of the error count:

a. Write 0x804C to address 0x9003C.
b. Poll address 0x90040 until bit 15 = 1; bits 19:16 represent bits 31:28 of the error count.
c. Write 0x004C to address 0x9003C.
d. Poll address 0x90040 until bit 15 = 0.

17. Finish checking the PRBS and BER test:
    a. Write 0xA041 to address 0x9003C.
    b. Poll address 0x90040 until bit 15 = 1.
    c. Write 0x2041 to address 0x9003C.
    d. Poll address 0x90040 until bit 15 = 0.

3.14.2.2.3. FGT Attribute Access Method Example 3

The following examples demonstrate the steps to perform polarity inversion for the FGT PMA.

To reverse the polarity of TX p and n pins:

1. Write 0x1A065 to address 0x9003C.
2. Poll address 0x90040 until bit 14 = 0 and bit 15 = 1.
3. Write 0x12065 to address 0x9003C.
4. Poll address 0x90040 until bit 14 = 0 and bit 15 = 0.

To reverse the polarity of RX p and n pins:

1. Write 0x1A066 to address 0x9003C.
2. Poll address 0x90040 until bit 14 = 0 and bit 15 = 1.
3. Write 0x12066 to address 0x9003C.
4. Poll address 0x90040 until bit 14 = 0 and bit 15 = 0.

To revert back the polarity of TX p and n pins:

1. Write 0x0A065 to address 0x9003C.
2. Poll address 0x90040 until bit 14 = 0 and bit 15 = 1.
3. Write 0x02065 to address 0x9003C.
4. Poll address 0x90040 until bit 14 = 0 and bit 15 = 0.

To revert back the polarity of RX p and n pins:

1. Write 0x0A066 to address 0x9003C.
2. Poll address 0x90040 until bit 14 = 0 and bit 15 = 1.
3. Write 0x02066 to address 0x9003C.
4. Poll address 0x90040 until bit 14 = 0 and bit 15 = 0.

4. Implementing the F-Tile Reference and System PLL Clocks Intel FPGA IP

The F-Tile Reference and System PLL Clocks Intel FPGA IP is required IP for F-tile PMA/FEC Direct PHY designs.

F-Tile Reference and System PLL Clocks Intel FPGA IP Overview

The F-Tile Reference and System PLL Clocks Intel FPGA IP performs three main functions, each described below:

- Configures the reference clock for FHT PMA:
  - Enable the FHT Common PLLs and select the reference clock source for FHT common PLL
  - Specify the FHT reference clock frequency
- Configures the reference clock for FGT PMA:
  - Enable FGT reference clocks and specify the reference clock frequency
  - To enable FGT CDR Output (RX recovered clock output)
- Configures the system PLL:
  - Enable system PLL and specify the mode
  - Specify the reference clock source and frequency for system PLL

4.1. IP Parameters

Table 89. F-Tile Reference and System PLL Clocks Intel FPGA IP Parameters

Parameter	Values	Description
System PLL #0		
Mode of system PLL	**Disabled**	Selects the mode of system PLL #0.
	User configuration	
	User PCIe-based configuration	
	ETHERNET_FREQ_805_156	
	ETHERNET_FREQ_805_312	
	ETHERNET_FREQ_805_322(41).	
	ETHERNET_FREQ_830_156	
	ETHERNET_FREQ_830_312	

continued...

(41) This mode is not currently supported

Parameter	Values	Description
	PCIE_FREQ_1000	• **Disabled**— system PLL not used. You must enable at least one system PLL. • **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. • **User configuration**—manually configures output frequency of system PLL and selects one of the feasible reference clocks. For use in non PCIe use cases when other Ethernet presets do not meet requirement. • **User PCIe-based configuration**— manually configures output frequency of system PLL and selects one of the feasible reference clocks. For use in PCIe use cases when PCIe presets do not meet requirement. The frequency number in the preset's label are abbreviated; they are not full precise frequency. Refer to the table below for full frequency. The default value is **ETHERNET_FREQ_805_156**.
	PCIE_FREQ_500	
	PCIE_FREQ_550	
	PCIE_FREQ_600	
	PCIE_FREQ_650	
	PCIE_FREQ_700	
	PCIE_FREQ_750	
	PCIE_FREQ_800	
	PCIE_FREQ_850	
	PCIE_FREQ_900	
	PCIE_FREQ_950	
Refclk source	**Refclk #0**	Selects the logical reference clock source for **system PLL #0**. The reference clock source can be shared with FGT PMA and other system PLLs. The default value is **Refclk #0**.
	Refclk #1	
	Refclk #2	
	Refclk #3	
	Refclk #4	
	Refclk #5	
	Refclk #6	
	Refclk #7	
	Refclk #8	
	Refclk #9	
Output frequency	31.25 to 1000 MHz	Specifies the output frequency of the **system PLL #0** 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. The default value is **805.6640625**.
System PLL #1		
Mode of system PLL	**Disabled**	Selects the mode of **system PLL #1**.
	User configuration	
	User PCIE-based configuration	
	ETHERNET_FREQ_805_156	
	ETHERNET_FREQ_805_312	
	ETHERNET_FREQ_805_322(41).	
	ETHERNET_FREQ_830_156	

continued...

Parameter	Values	Description
	ETHERNET_FREQ_830_312	• **Disabled**— system PLL not used. You must enable at least one system PLL. • **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. • **User configuration**—manually configures output frequency of system PLL and selects one of the feasible reference clocks. For use in non PCIe use cases when other Ethernet presets do not meet requirement. • **User PCIe-based configuration**— manually configures output frequency of system PLL and selects one of the feasible reference clocks. For use in PCIe use cases when PCIe presets do not meet requirement. The default value is **Disabled**.
	ETHERNET_FREQ_830_312	
	PCIE_FREQ_1000	
	PCIE_FREQ_500	
	PCIE_FREQ_550	
	PCIE_FREQ_600	
	PCIE_FREQ_650	
	PCIE_FREQ_700	
	PCIE_FREQ_750	
	PCIE_FREQ_800	
	PCIE_FREQ_850	
	PCIE_FREQ_900	
	PCIE_FREQ_950	
Refclk source	**Refclk #0**	Selects the logical reference clock source for **system PLL #1**. The reference clock source can be shared with FGT PMA and other system PLLs.
	Refclk #1	
	Refclk #2	
	Refclk #3	
	Refclk #4	
	Refclk #5	
	Refclk #6	
	Refclk #7	
	Refclk #8	
	Refclk #9	
Output frequency	31.25 to 1000 MHz	Specifies the output frequency of the **system PLL #1** in MHz. In background, the algorithm calculates the legal reference clock frequencies for that clock output frequency. For correct calculation, must specify the exact frequency with decimal points.
System PLL #2		
Mode of system PLL	**Disabled**	Selects the mode of **system PLL #2**.
	User configuration	
	User PCIE-based configuration	
	ETHERNET_FREQ_805_156	
	ETHERNET_FREQ_805_312	
	ETHERNET_FREQ_805_322(41).	
	ETHERNET_FREQ_830_156	
	ETHERNET_FREQ_830_312	

continued...

Parameter	Values	Description
	ETHERNET_FREQ_830_312	• **Disabled**— system PLL not used. • **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. • **User configuration**—manually configures output frequency of system PLL and selects one of the feasible reference clocks. For use in non PCIe use cases when other Ethernet presets do not meet requirement. • **User PCIe-based configuration**— manually configures output frequency of system PLL and selects one of the feasible reference clocks. For use in PCIe use cases when PCIe presets do not meet requirement. The frequency number in the Preset's label are abbreviated, they are not full precise frequency.
	PCIE_FREQ_1000	
	PCIE_FREQ_500	
	PCIE_FREQ_550	
	PCIE_FREQ_600	
	PCIE_FREQ_650	
	PCIE_FREQ_700	
	PCIE_FREQ_750	
	PCIE_FREQ_800	
	PCIE_FREQ_850	
	PCIE_FREQ_900	
	PCIE_FREQ_950	
Refclk source	**Refclk #0**	Selects the logical reference clock source for **system PLL #2.** The reference clock source can be shared with FGT PMA and other system PLLs.
	Refclk #1	
	Refclk #2	
	Refclk #3	
	Refclk #4	
	Refclk #5	
	Refclk #6	
	Refclk #7	
	Refclk #8	
	Refclk #9	
Output Frequency	31.25 to 1000 MHz	Specifies the output frequency of the **system PLL #2** in MHz. In background, the algorithm calculates the legal reference clock frequencies for that clock output frequency. For correct calculation, must specify the exact frequency with decimal points.
FHT Common PLL		
Controller source	**Auto**, **CommonPLL A**, **CommonPLL B**	If both common PLLs are enabled, this selection specifies the common PLL that drives the FHT microcontroller. The reference clock that drives this common PLL must be present and stable throughout F-tile operation.
FHT Common PLL A		
Enable FHT Common PLL A	**On/Off**	Enable/Disable FHT common PLL A. When enabled, must provide FHT reference clock source and frequency. The default value is **Off**.
FHT refclk source	**FHT Refclk #0**	Specifies the logical reference clock source for FHT common PLL A. The default value is **FHT Refclk #0**.
	FHT Refclk #1	
FHT Common PLL B		

continued...

Parameter	Values	Description
Enable FHT Common PLL B	**On/Off**	When enabled, must provide FHT reference clock source and frequency. The default value is **Off**.
FHT refclk source	**FHT Refclk #0** **FHT Refclk #1**	Specifies the logical reference clock source for FHT common PLL B. The default value is **FHT Refclk #0**.
Reference clock(s)		
FGT/System PLL		
Enable Refclk #0 for FGT PMA	**On/Off**	Enables logical reference clock #0 for FGT PMA. This reference clock can also be shared by system PLL. The default value is **Off**.
Refclk frequency #0	25 to 380 MHz	Specifies the reference clock #0 frequency. Range is: • 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA. When reference clock #0 is shared between FGT PMA and system PLL, the parameter editor allows selection of calculated legal frequencies. The default value is **156.25 MHz**.
Refclk #0 is active at and after device configuration	**On/Off**	When **On**, you must provide the reference clock #0 which is free running and stable at and after device programming time. When **Off**, the reference clock can be inactive at device programming time, or can go down during device operation. Special handling of `en_refclk_fgt_0` signal is required. The default value is **On**.
Export Refclk #0 for use in user logic	**On/Off**	Allow FGT reference clock #0 to be used in user logic. The default value is **Off**.
Enable Refclk #1 for FGT PMA	**On/Off**	Enables logical reference clock #1 for FGT PMA. This reference clock can also be shared by system PLL. The default value is **Off**.
Refclk frequency #1	25 to 380 MHz	Specifies the reference clock #1 frequency. Range is: • 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA When reference clock #1 is shared between FGT PMA and system PLL, reference clock frequency changes to drop-down of calculated legal frequencies and user must select one from drop-down. The default value is **Disabled**.
Refclk #1 is active at and after device configuration	**On/Off**	When **On**, you must provide the reference clock #1 which is free running and stable at and after device programming time. When **Off**, the reference clock can be inactive at device programming time, or can go down during device operation. Special handling of `en_refclk_fgt_1` signal is required. The default value is **On**.

continued...

Parameter	Values	Description
Export Refclk #1 for use in user logic	**On/Off**	Allow FGT reference clock #1 to be used in user logic. The default value is **Off**.
Enable Refclk #2 for FGT PMA	**On/Off**	Enable logical reference clock #2 for FGT PMA. This reference clock can also be shared by system PLL. The default value is **Off**.
Refclk frequency #2	25 to 380 MHz	Specifies the reference clock #2 frequency. Range is: • 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA When reference clock #2 is shared between FGT PMA and system PLL, the parameter editor allows selection of calculated legal frequencies. The default value is **Disabled**.
Refclk #2 is active at and after device configuration	**On/Off**	When **On**, you must provide the reference clock #2 which is free running and stable at and after device programming time. When **Off**, the reference clock can be inactive at device programming time, or can go down during device operation. Special handling of `en_refclk_fgt_2` signal is required. The default value is **On**.
Export Refclk #2 for use in user logic	**On/Off**	Allow FGT reference clock #2 to be used in user logic. The default value is **Off**.
Enable Refclk #3 for FGT PMA	**On/Off**	Enable logical reference clock #3 for FGT PMA. This reference clock can also be shared by system PLL. The default value is **Off**.
Refclk frequency #3	25 to 380 MHz	Specifies the reference clock #3 frequency. Range is: • 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA When reference clock #3 is shared between FGT PMA and system PLL, the parameter editor allows selection of calculated legal frequencies. The default value is **Disabled**.
Refclk #3 is active at and after device configuration	**On/Off**	When **On**, you must provide the reference clock #3 which is free running and stable at and after device programming time. When **Off**, the reference clock can be inactive at device programming time, or can go down during device operation. Special handling of `en_refclk_fgt_3` signal is required. The default value is **On**.
Export Refclk #3 for use in user logic	**On/Off**	Allow FGT reference clock #3 to be used in user logic. The default value is **Off**.
Enable Refclk #4 for FGT PMA	**On/Off**	Enable logical reference clock #4 for FGT PMA. This reference clock can also be shared by system PLL. The default value is **Off**.
Refclk frequency #4	25 to 380 MHz	Specifies the reference clock #4 frequency. Range is:

continued...

Parameter	Values	Description
		• 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA. When reference clock #4 is shared between FGT PMA and system PLL, , the parameter editor allows selection of calculated legal frequencies. The default value is **Disabled**.
Refclk #4 is active at and after device configuration	**On/Off**	When **On**, you must provide the reference clock #4 which is free running and stable at and after device programming time. When **Off**, the reference clock can be inactive at device programming time, or can go down during device operation. Special handling of `en_refclk_fgt_4` signal is required. The default value is **On**.
Export Refclk #4 for use in user logic	**On/Off**	Allow FGT reference clock #4 to be used in user logic. The default value is **Off**.
Enable Refclk #5 for FGT PMA	**On/Off**	Enable logical reference clock #5 for FGT PMA. This reference clock can also be shared by system PLL. The default value is **Off**.
Refclk frequency #5	25 to 380 MHz	Specifies the reference clock #5 frequency. Range is: • 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA. When reference clock #5 is shared between FGT PMA and system PLL, , the parameter editor allows selection of calculated legal frequencies. The default value is **Disabled**.
Refclk #5 is active at and after device configuration	**On/Off**	When **On**, you must provide the reference clock #5 which is free running and stable at and after device programming time. When **Off**, the reference clock can be inactive at device programming time, or can go down during device operation. Special handling of `en_refclk_fgt_5` signal is required. The default value is **On**.
Export Refclk #5 for use in user logic	**On/Off**	Allow FGT reference clock #5 to be used in user logic. The default value is **Off**.
Enable Refclk #6 for FGT PMA	**On/Off**	Enable logical reference clock #6 for FGT PMA. This reference clock can also be shared by system PLL. The default value is **Off**.
Refclk frequency #6	25 to 380 MHz	Specifies the reference clock #6 frequency. Range is: • 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA.

continued...

Parameter	Values	Description
		When reference clock #6 is shared between FGT PMA and system PLL, , the parameter editor allows selection of calculated legal frequencies. The default value is **Disabled**.
Refclk #6 is active at and after device configuration	**On/Off**	When **On**, you must provide the reference clock #6 which is free running and stable at and after device programming time. When **Off**, the reference clock can be inactive at device programming time, or can go down during device operation. Special handling of `en_refclk_fgt_6` signal is required. The default value is **On**.
Export Refclk #6 for use in user logic	**On/Off**	Allow FGT reference clock #6 to be used in user logic. The default value is **Off**.
Enable Refclk #7 for FGT PMA	**On/Off**	Enable logical reference clock #7 for FGT PMA. This reference clock can also be shared by system PLL. The default value is **Off**.
Refclk frequency #7	25 to 380 MHz	Specifies the reference clock #7 frequency. Range is: • 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA. When reference clock #7 is shared between FGT PMA and system PLL, the parameter editor allows selection of calculated legal frequencies. The default value is **Disabled**.
Refclk #7 is active at and after device configuration	**On/Off**	When **On**, you must provide the reference clock #7 which is free running and stable at and after device programming time. When **Off**, the reference clock can be inactive at device programming time, or can go down during device operation. Special handling of `en_refclk_fgt_7` signal is required. The default value is **On**.
Export Refclk #7 for use in user logic	**On/Off**	Allow FGT reference clock #7 to be used in user logic. The default value is **Off**.
Enable Refclk #8 for FGT PMA	**On/Off**	Enable logical reference clock #8 for FGT PMA. This reference clock can also be shared by system PLL. The default value is **Off**.
Refclk frequency #8	25 to 380 MHz	Specifies the reference clock #8 frequency. Range is: • 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA. When reference clock #8 is shared between FGT PMA and system PLL, the parameter editor allows selection of calculated legal frequencies. . The default value is **Disabled**.
Refclk #8 is active at and after device configuration	**On/Off**	When **On**, you must provide the reference clock #8 which is free running and stable at and after device programming time.

continued...

Parameter	Values	Description
		When **Off**, the reference clock can be inactive at device programming time, or can go down during device operation. Special handling of `en_refclk_fgt_8` signal is required. The default value is **On**.
Export Refclk #8 for use in user logic	**On/Off**	Allow FGT reference clock #8 to be used in user logic. The default value is **Off**.
Enable Refclk #9 for FGT PMA	**On/Off**	Enable logical reference clock #9 for FGT PMA. This reference clock can also be shared by system PLL. The default value is **Off**.
Refclk frequency #9	25 to 380 MHz	Specifies the reference clock #9 frequency. Range is: • 25 – 380 MHz when reference clock is configured for FGT PMA. (If using HDMI protocol, use only 25 – 100 MHz) • 100 – 380 MHz when reference clock is configured for System PLL or shared with system PLL and FGT PMA. When reference clock #9 is shared between FGT PMA and system PLL, the parameter editor allows selection of calculated legal frequencies. . The default value is **Disabled**.
Refclk #9 is active at and after device configuration	**On/Off**	When **On**, you must provide the reference clock #9 which is free running and stable at and after device programming time. When **Off**, the reference clock can be inactive at device programming time, or can go down during device operation. Special handling of `en_refclk_fgt_9` signal is required. The default value is **On**.
Export Refclk #9 for use in user logic	**On/Off**	Allow FGT reference clock #9 to be used in user logic. The default value is **Off**.
FGT CDR Clock-out(s)		
Enable FGT CDR Output #0	**On/Off**	Enables logical FGT CDR clock output #0. This must be enabled to configure FGT reference clock as a CDR clock output. The default value is **Off**.
Enable FGT CDR Output #1	**On/Off**	Enables logical FGT CDR clock output #1. This must be enabled to configure FGT reference clock as a CDR clock output. The default value is **Off**.
FHT Reference clock(s)		
FHT Refclk frequency #0	100 to 200 MHz	Specifies the FHT reference clock #0 frequency in MHz.
FHT Refclk frequency #1	100 to 200 MHz	Specifies the FHT reference clock #1 frequency in MHz.

4.2. IP Port List

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

Table 90. F-Tile Reference and System PLL Clocks Intel FPGA IP Port List

Refer to Port Connection Guidelines between F-Tile Reference and System PLL Clocks Intel FPGA IP and F-Tile PMA/FEC Direct PHY Intel FPGA IP for recommended connections.

Port Name	Direction	Description
FHT		
in_refclk_fht_i	Input	FHT reference clock input port. Must be mapped to device reference clock pin. Maximum of 2 (i = 0 to 1) ports of this type.
out_fht_cmmpll_clk_i	Output	FHT common PLL output port. Must be connected to protocol IPs, connected to FHT building-block. There can be a maximum of 2(i = 0 to 1) ports of this type.
FGT and System PLL		
in_refclk_fgt_i	Input	FGT and system PLL reference clock input port. Must be mapped to device reference clock pin. This reference clock port can be connected to FGT PMA, system PLL or both. There can be a maximum of 10 (i = 0 to 9) ports of this type.
avmm_clk	Input	Avalon memory-mapped interface clock. This port is only available when at least one of **Refclk #i is active at and after device configuration** is set as **Off**. Intel recommends 100 to 250 MHz for this clock.
avmm_reset	Input	Avalon memory-mapped interface reset. This port is only available when at least one of **Refclk #i is active at and after device configuration** is set as **Off**.
refclock_ready [2:0]	Input	System PLL reference clock status control signal. This port is only available when all the enabled system PLL's corresponding **Refclk #i is active at and after device configuration** are set as **Off**. • bit[0] is used to control system PLL #0 reference clock. • bit[1] is used to control system PLL #1 reference clock. • bit[2] is used to control system PLL #2 reference clock. When system PLL #i is disabled, bit[i] can be any value and does not matter. When system PLL #i is enabled, after the reference clock is available, you must assert bit[i].
FGT		
out_refclk_fgt_i	Output	FGT Refclk output port. Must be connected to protocol IPs, connected to FGT building-block. There can be a maximum of 10 (i = 0 to 9) ports of this type.
in_cdrclk_i	Input	Input port for FGT reference clock configured as CDR output. This must be connected to protocol IP output CDR port. There can be a maximum of 2 (i = 0 to 1) ports of this type.
out_cdrclk_i	Output	Output port for FGT reference clock configured as CDR output. This must be connected to one of two FGT reference clock pins that can be configured as CDR outputs. You must specify the location assignment in the Intel Quartus Prime Pro Edition software qsf settings file for correct functionality. There can be a maximum of 2 (i = 0 to 1) ports of this type.

continued...

Port Name	Direction	Description
en_refclk_fgt_i[1:0]	Input	FGT reference clock status control signal. This port is only available when the corresponding **Refclk #i is active at and after device configuration** set as **Off**. There can be a maximum of 10 (i = 0 to 9) ports of this type. • 2'b00: Disable Refclk #i • 2'b01: Enable Refclk #i • 2'b11: Monitoring Refclk #i When the Refclk #i is inactive, you should drive en_refclk_fgt_i to 2'b00. When the Refclk #i is active, you should drive en_refclk_fgt_i to 2'b01. When the device starts operating normally, you should drive en_refclk_fgt_i to 2'b11 to monitor the Refclk #i status.
refclk_fgt_enabled_i	Output	FGT reference clock status signal. This port is only available when the corresponding **Refclk #i is active at and after device configuration** is set to **Off**. The signal is only valid when the en_refclk_fgt_i set to 2'b11. • 1'b0: Refclk #i is inactive • 1'b1: Refclk #i is active You can use the status of this signal to control the en_refclk_fgt_i signal.
out_coreclk_i	Output	FGT reference clock output port for user logic. This port is only available when the corresponding **Export Refclk #i for use in user logic** is set to **On**. There can be a maximum of 10 (i = 0 to 9) ports of this type.
System PLL		
out_systempll_clk_i	Output	Output port of system PLL. This must be connected to system PLL clock input of protocol IP. There can be a maximum of 3 (i = 0 to 2) ports of this type.
out_systempll_synthlock_i	Output	System PLL lock status port which indicates if system PLL is locked to incoming reference clock. There can be a maximum of 3 (i = 0 to 2) ports of this type. You can use this port as a status or debug signal.

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

Table 91. Preset Reference Clock and Output Frequencies

Mode of System PLL - System PLL	Reference Clock (MHz)	Output Frequency (MHz)
ETHERNET_FREQ_805_156	156.25	805.6640625
ETHERNET_FREQ_805_312	312.5	805.6640625
ETHERNET_FREQ_805_322[42].	322.265625	805.6640625
ETHERNET_FREQ_830_156	156.25	830.078125
ETHERNET_FREQ_830_312	312.5	830.078125
PCIE_FREQ_1000	100	1000
PCIE_FREQ_950	100	950

continued...

[42] This mode is not currently supported

Mode of System PLL - System PLL	Reference Clock (MHz)	Output Frequency (MHz)
PCIE_FREQ_900	100	900
PCIE_FREQ_850	100	850
PCIE_FREQ_800	100	800
PCIE_FREQ_750	100	750
PCIE_FREQ_700	100	700
PCIE_FREQ_650	100	650
PCIE_FREQ_600	100	600
PCIE_FREQ_550	100	550
PCIE_FREQ_500	100	500

Table 92. Port Connection Guidelines between F-Tile Reference and System PLL Clocks Intel FPGA IP and F-Tile PMA/FEC Direct PHY Intel FPGA IP

F-Tile Reference and System PLL Clocks Intel FPGA IP	F-Tile PMA/FEC Direct PHY Intel FPGA IP
System PLL	
`out_systempll_clk`	`system_pll_clk_link`
FGT	
`out_refclk_fgt`	`tx_pll_refclk_link`, `rx_cdr_refclk_link`
`in_cdrclk`	`rx_cdr_divclk_link`
FHT	
`out_fht_cmmpll_clk`	`tx_pll_refclk_link`, `rx_cdr_refclk_link`

4.4. Guidelines for F-Tile Reference and System PLL Clocks Intel FPGA IP Usage

You must adhere to the following guidelines to correctly use the F-Tile Reference and System PLL Clocks Intel FPGA IP:

- The F-Tile Reference and System PLL Clocks Intel FPGA IP must always connect to the F-Tile PMA/FEC Direct PHY Intel FPGA IP or protocol IPs. You cannot compile or simulate the F-Tile Reference and System PLL Clocks Intel FPGA IP as a standalone IP.
- 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.
- You must connect the reference clock and system PLL output ports of F-Tile Reference and System PLL Clocks Intel FPGA IP to input of F-Tile PMA/FEC Direct PHY Intel FPGA IP as shown in Port Connection Guidelines between F-Tile Reference and System PLL Clocks Intel FPGA IP and F-Tile PMA/FEC Direct PHY Intel FPGA IP or protocol IPs.

- You must ensure the reference clock and system PLL frequencies specified in F-Tile Reference and System PLL Clocks Intel FPGA IP match reference clock and system PLL frequencies specified in F-Tile PMA/FEC Direct PHY Intel FPGA IP or protocol IPs. Any mismatch in frequency results in Intel Quartus Prime Pro Edition software **Support-Logic Generation** failure.
- You must enable at least one system PLL per F-tile because this is a requirement for F-tile configuration to pass successfully. Enabling at least one system PLL is required even when the data path is using PMA clocking mode. If your design has one system PLL enabled to be used for system PLL clocking, you do not need a separate system PLL for F-tile configuration. When you use the system PLL only for F-tile configuration (that is, when all lanes use the PMA clocking mode) the following guidelines apply:
  - You must enable **System PLL #0**. If you enable **System PLL #1** or **System PLL #2**, the Intel Quartus Prime Pro Edition software **Support-Logic Generation** step fails.
  - The system PLL output must be unconnected. This is the only exception where you can leave the system PLL output unconnected. In all other scenarios you must always connect the system PLL output to F-Tile PMA/FEC Direct PHY Intel FPGA IP or protocol IPs.
  - If you are not using the FGT PMA, the reference clock to system PLL connection is not necessary (that is, you do not need to connect the reference clock); however, if you connect a reference clock, the configuration completes faster.
  - If you are using the FGT PMA, the reference clock to system PLL connection is necessary (that is, you must connect the reference clock).
- When you instantiate multiple interfaces or protocol-based IP cores within a single F-tile, you must use only one instance of the F-Tile Reference and System PLL Clocks Intel FPGA IP to configure the following:
  - All reference clocks for the FGT PMA (up to 10) and the FHT PMA (up to 2) that are required to implement those multiple interfaces within a single F-tile.
  - All FHT common PLLs (up to 2) that are required to implement those multiple interfaces within a single F-tile.
  - All system PLLs (up to 3) that are required to implement those multiple interfaces within a single F-tile.
  - All reference clocks for system PLLs (up to 8, shared with the FGT PMA) that are required to implement those multiple interfaces within a single F-tile.

When you design multiple interfaces or protocols based IP cores within a single F-tile, you can only use three system PLLs. For example, you can use one system PLL for PCIe and two for Ethernet and other protocols. However, there are other use cases where you can use all three for various interfaces within the Ethernet and PMA direct digital blocks. As there are only three system PLLs, multiple interfaces or protocol based IP cores with different line rates may have to share a system PLL. While sharing a system PLL, the interface with the highest line rate determines the system PLL frequency, and the interfaces with the lower line rates must be overclocked.

All reference clock, system PLL and common PLL selection in the IP parameter editor are logical. The `.qsf` assignments map these logical selection to physical resources.

- Although system PLL reference clock source lists ten reference clocks (reference clock #0 to #9), only eight physical reference clocks can clock the system PLL. For example, you could select reference clock #10 as the system PLL reference clock source, but this must be physically mapped to FGT/System PLL reference clock location 0 to 7 by specifying the `.qsf` assignments.
- When you enable the FGT CDR Output (RX recovered clock output), you must physically map the corresponding FGT PMA to FGT Quad 2 or 3, and you must physically map the FGT CDR Output (RX recovered clock output) to the FGT reference clock location 8 or 9 (configured as output).
- The total number of FGT/system PLL reference clocks and FGT CDR clock out that are enabled must not exceed 10.

Related Information

Enabling Custom Cadence Generation Ports and Logic on page 211

4.5. Guidelines for Refclk #i is Active At and After Device Configuration

The **Refclk #i is active at and after device configuration** feature is only supported by Intel Agilex 7 F-tile devices mentioned in *Appendix A.1*. In designs that target other OPN devices, you must set the **Refclk #i is active at and after device configuration** to **On**, and you must provide stable reference clocks for the system PLLs and FGT at and after device configuration time.

- For Refclk #i that is being used by system PLL only, refer to *Guidelines for System PLL Reference Clock*.
- For Refclk #i that is being used by FGT PMA only, refer to *Guidelines for FGT Reference Clock*.
- For Refclk #i that is being used by both system PLL and FGT PMA, refer to *Guidelines for System PLL Reference Clock* and *Guidelines for FGT Reference Clock*.

4.5.1. Guidelines for System PLL Reference Clock

For reference clock, Refclk #i (i = 0 to 9) that are being used by system PLL #n (n = 0, 1, 2):

- When the parameter **Refclk #i is active at and after device configuration** is set to **On**, the refclk #i must be active at and after device configuration time.
- When the parameter **Refclk #i is active at and after device configuration** is set to **Off**, the refclk #i can be active after device configuration time. After the refclk #i is active, you need to assert `refclock_ready[n]` signal to indicate system PLL #n reference clock is ready.
- Once refclk #i is active, it must be stable and present throughout the device operation and must not go down.

All refclk #i that are being used by system PLLs must have the **Refclk #i is active at and after device configuration** parameter set to the same value. Only the following two cases are supported.

- All system PLL reference clocks have the **Refclk #i is active at and after device configuration** parameter set to **On**.
- Or all system PLLs reference clocks have the **Refclk #i is active at and after device configuration** parameter set to **Off**.

Note: If refclk #0 and refclk #1 are both used by system PLLs, where **Refclk #0 is active at and after device configuration** parameter is set to **On**, and **Refclk #1 is active at and after device configuration** parameter is set to **Off**, the F-Tile Reference and System PLL Clocks Intel FPGA IP does not expose the `refclock_ready` signal.

When **Refclk #i is active at and after device configuration** parameter is set to **Off**, an internal clock is used to calibrate and configure the FPGA device. Due to the low frequency of the internal clock, the calibration and configuration takes longer to finish. In addition, after the system PLL #n reference clock is ready, you must assert `refclock_ready[n]`. This flow may not meet the link up requirements for some IP protocols. You must make sure your design application is compatible with this flow. Intel recommends supplying a stable and running system PLL reference clock at device configuration, and enabling the **Refclk #i is active at and after device configuration** parameter.

For PCIe interfaces that require compliance to PCIe link training specifications, the reference clock to the system PLL must be available and stable before device configuration begins. You must set the **Refclk #i is active at and after device configuration** parameter in the F-Tile Reference and System PLL Clocks Intel FPGA IP to **On** and drive the reference clock from an independent and free running clock source. Alternately, if the reference clock from the PCIe link is guaranteed to be available before device configuration begins, you may use it to drive the system PLL. Once the PCIe link reference clock is alive, it must never go down.

4.5.2. Guidelines for FGT Reference Clock

For Refclk #i (i = 0 to 9) that are being used by the FGT PMA only:

- When the parameter **Refclk #i is active at and after device configuration** is set to **On**, the refclk #i must be active at and after device configuration time.
- When the parameter **Refclk #i is active at and after device configuration** is set to **Off**, the refclk #i can be inactive at any time. You must drive `en_refclk_fgt_i` to `2'b00` when refclk #i is inactive, and drive `en_refclk_fgt_i` to `2'b01` when refclk #i is active.

5. F-tile PMA/FEC Direct PHY Design Implementation

This chapter describes the IP parameterization, PHY IP connections, simulation, and tile placement planning for a F-tile PMA/FEC Direct PHY design. The design implements two 25.78125 Gbps NRZ PMA Direct FGT lanes, with a throughput of 51.5625 Gbps, and with system PLL datapath clocking mode.

5.1. Implementing the F-tile PMA/FEC Direct PHY Design

Note: This topic through Connecting the F-tile PMA/FEC Direct PHY Design IP on page 212 explains how to implement a PMA/FEC Direct design by instantiating and connecting the necessary IP components. You can alternatively use the Example Design options that Example Design Generation on page 107 describes to implement an example design.

This design implementation requires the following IP available from the Intel Quartus Prime Pro Edition software IP Catalog:

- F-Tile PMA/FEC Direct PHY Intel FPGA IP
- F-Tile Reference and System PLL Clocks Intel FPGA IP

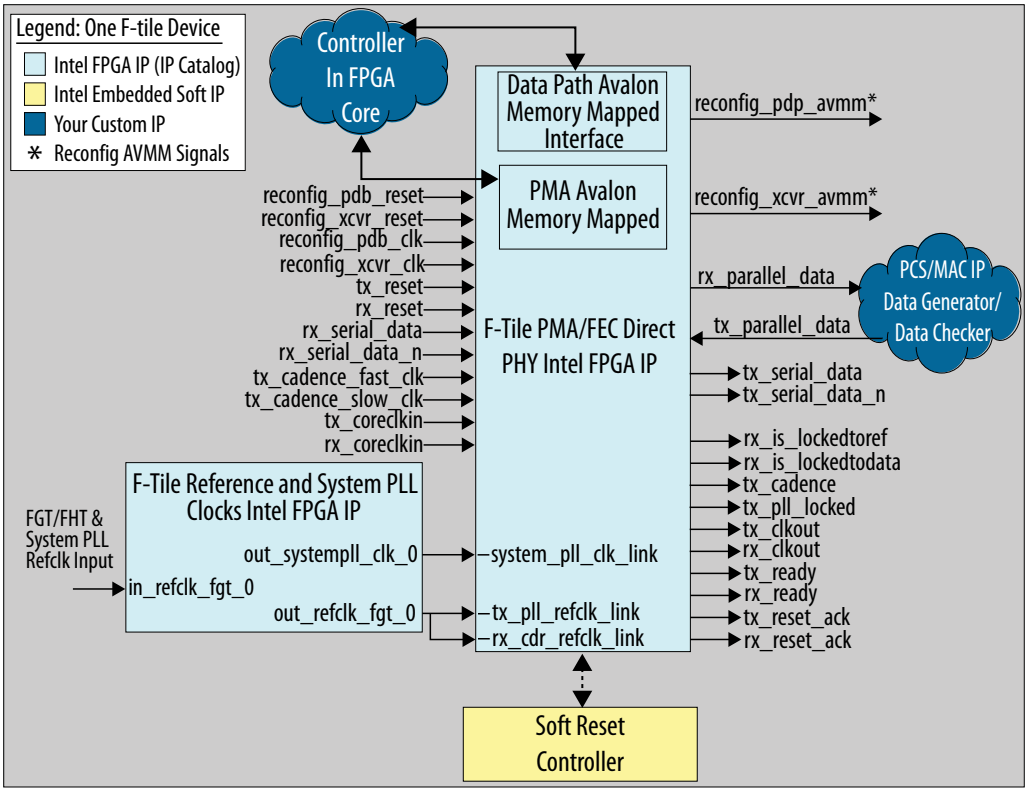
The F-Tile PMA/FEC Direct PHY Intel FPGA IP is the primary IP component for PMA and FEC direct implementation. This IP provides direct access to the F-tile PMA block features for both FGT and FHT.

To customize and instantiate the IP for your protocol implementation, you specify parameter values for the F-Tile 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. You use these ports to connect the F-Tile PMA/FEC Direct PHY Intel FPGA IP to other IP components in your design. These include connections to the respective reference clock pins and system PLL clock outputs from the F-Tile Reference and System PLL Clocks Intel FPGA IP, TX and RX parallel data ports, as well as TX and RX serial data pins.

F-tile PMA/FEC Direct PHY Design IP Connections shows the connections between the IP design blocks required for the F-tile PMA/FEC Direct PHY design. The diagram illustrates the connections between the F-Tile Reference and System PLL Clocks Intel FPGA IP, the Soft Reset Controller (that instantiates automatically after running **Design Analysis**), and the user-provided MAC/PCS IP core into the parallel data bus to the F-Tile PMA/FEC Direct PHY Intel FPGA IP.

Figure 94. F-tile PMA/FEC Direct PHY Design IP Connections



Note: (*)For full port lists, including the reconfig_pdp_avmm and reconfig_xcvr_avmm signals, refer to PMA Avalon Memory Mapped Interface Signals on page 118 and Datapath Avalon Memory Mapped Interface Signals on page 120.

The following topics describe PHY IP parameterization, connection, simulation, and tile placement planning for the design:

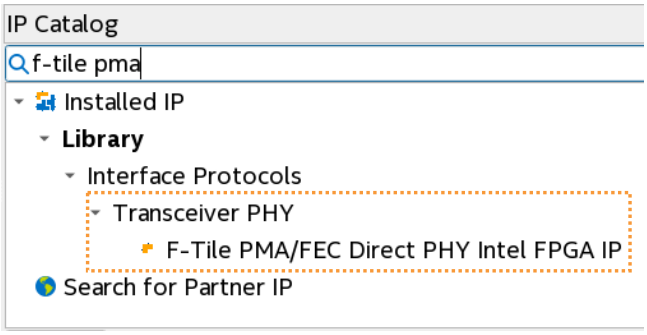
1. Instantiating the F-Tile PMA/FEC Direct PHY Intel FPGA IP on page 201
2. Instantiating the F-Tile Reference and System PLL Clocks Intel FPGA IP on page 209
3. Enabling Custom Cadence Generation Ports and Logic on page 211
4. Connecting the F-tile PMA/FEC Direct PHY Design IP on page 212
5. Simulating the F-Tile PMA/FEC Direct PHY Design on page 212
6. F-tile Interface Planning on page 215

5.2. Instantiating the F-Tile PMA/FEC Direct PHY Intel FPGA IP

To instantiate the F-Tile PMA/FEC Direct PHY Intel FPGA IP:

1. Specify the target device family, click **Assignments ➤ Device**, and then select **Agilex AGIB027R29A2E2V**.
2. If IP catalog is not already open, click **View ➤ IP Catalog** in the Intel Quartus Prime software.
3. In the IP Catalog search field, type `f-tile pma`, and double-click the **F-Tile PMA/FEC Direct PHY Intel FPGA IP**.

Figure 95. F-Tile PMA/FEC Direct PHY Intel FPGA IP in IP Catalog

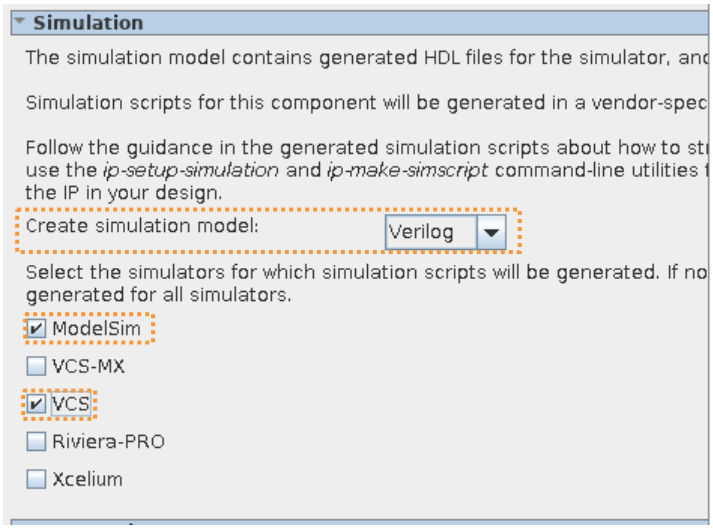


4. In the parameter editor, specify optional values to configure the F-Tile PMA/FEC Direct PHY Intel FPGA IP for your protocol implementation:
   - Setting General and Common Datapath Options on page 202
   - Setting TX Datapath Options on page 204
   - Setting RX Datapath Options on page 206

   You can optionally specify the **FGT_NRZ_50G_2_PMA_Lanes_Custom_Cadence_ED** in the collection of **Presets** to apply those default parameter values. During parameterization, instantiate the PMA direct channel. The available parameter editor options reflect your channel requirements.
5. When parameterization is complete, click the **Generate HDL** button in the parameter editor to generate the IP instance and supporting files. Under **Simulation**, select **Verilog** and either **VCS** or **ModelSim** for **Create simulation model**.(43)

(43) The current Intel Quartus Prime software version supports only VCS or ModelSim for F-tile simulation.

Figure 96. Simulation Options



6. Click the **Generate** 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 that generates with the IP instance includes all the available ports for your configuration. Use these ports to connect the F-Tile PMA/FEC Direct PHY Intel FPGA IP to other IP cores in your design, as Connecting the F-tile PMA/FEC Direct PHY Design IP on page 212 describes.

5.2.1. Setting General and Common Datapath Options

The following parameter values define an instance of F-Tile PMA/FEC Direct PHY Intel FPGA IP with the following properties:

- Two FGT PMA operating at duplex mode, NRZ modulation, 32-bit data width
- Two PMA lanes with data rate at 25.78125Gbps (supports 51.5625Gbps link)
- System PLL datapath clocking mode and the output frequency

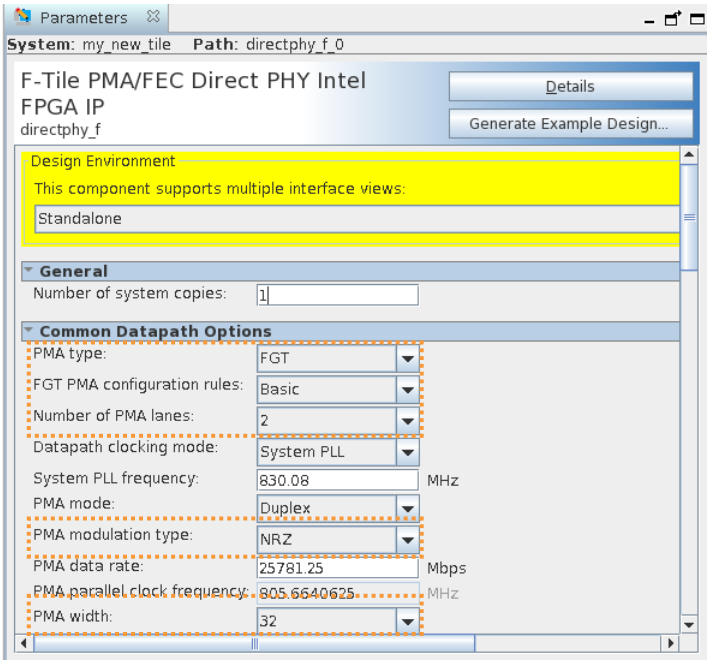
Table 93. General and Common Datapath Options

Parameter	Parameter Value
PMA type	**FGT**
Number of PMA lanes	**2**
FGT PMA configuration rules	**Basic**
PMA modulation type	**NRZ**
PMA width	**32**

This parameterization uses two FGT PMA lanes. There are two EMIB, connecting to two PMA lanes, to form two streams of PMA direct datapath. The following is the bonding configuration for this example:

- PMA bonding is disabled because the **PMA width** is 32-bit.
- System bonding is enabled so two PMA lanes bond to form the 51.625Gbps link.

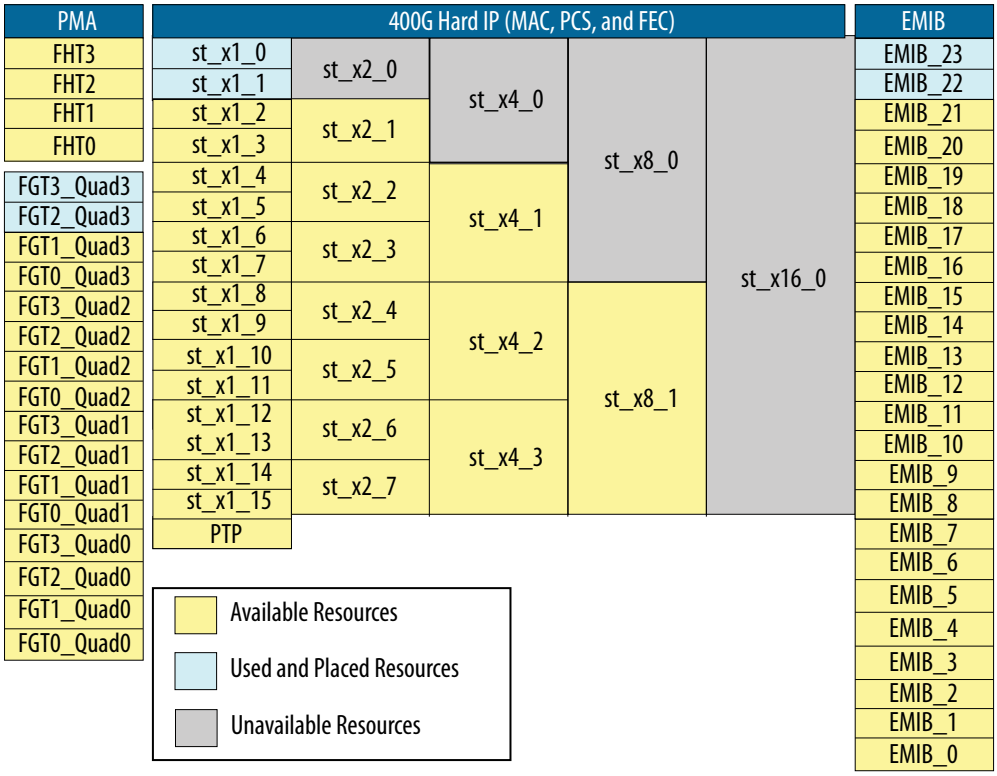
Figure 97. General and Common Datapath Options



400G Hard IP Fracture Resources Used and Placed by the Design shows the assignment of the PMA physical lanes to FGT3_Quad3 and FGT2_Quad3 location. This physical location of the PMA lanes uses the resources highlighted in (light blue) in the diagram. These resources are:

- PMA lanes placed in FGT3_Quad3 and FGT2_Quad3
- Fracture st_x1_0 and st_x1_1
- EMIB_23 and EMIB_22

Figure 98. 400G Hard IP Fracture Resources Used and Placed by the Design



5.2.2. Setting TX Datapath Options

Specify options for the following on the F-Tile PMA/FEC Direct PHY Intel FPGA IP parameter editor **TX Datapath Options** tab:

- TX FGT PMA
- TX FGT PLL
- TX datapath FIFO modes

The design specifies the following **TX Datapath Options**:

Table 94. TX FGT PLL Options

Parameter	Parameter Value
TX FGT PLL reference clock frequency	Select **156.25MHz**. The **TX FGT PLL reference clock frequency** must match the reference clock frequency that the F-Tile Reference and System PLL Clocks Intel FPGA IP specifies, as TX FGT PLL Settings shows. To connect the `out_refclk_fgt_0` to this IP, refer to Connecting the F-tile PMA/FEC Direct PHY Design IP on page 212

Figure 99. TX FGT PLL Settings

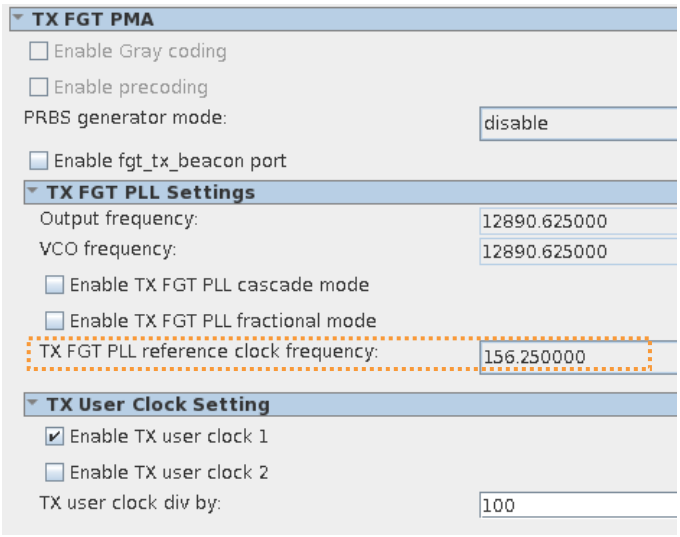
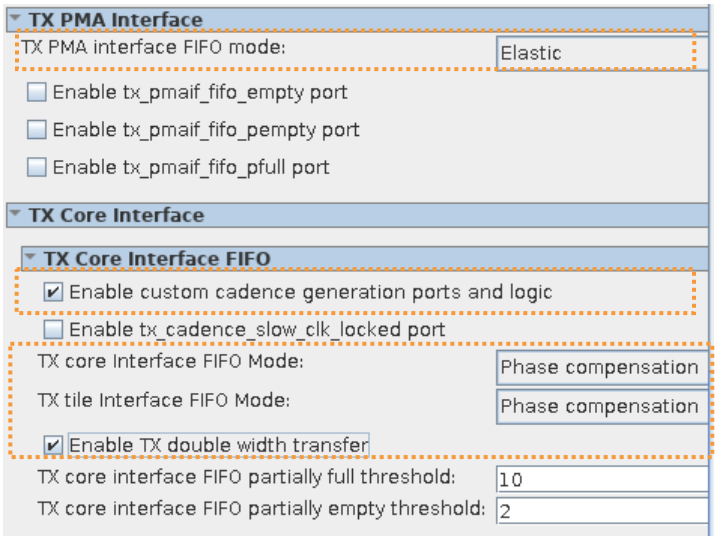


Table 95. TX PMA Interface Options

Parameter	Parameter Value
TX PMA interface FIFO mode	**Elastic**
Enable custom cadence generation ports and logic	Generates the `tx_cadence` port that you can use to assert and de-assert the PMA data valid bit. This option is needed because the system PLL frequency is greater than the PMA clock frequency in this design. Refer to Custom Cadence Generation Ports and Logic on page 145.
TX core Interface FIFO Mode	**Phase Compensation**
TX tile FIFO Interface FIFO Mode	**Phase Compensation**
Enable TX double width transfer	On. When On, you must drive the `tx_clkout` source with **Sys PLL Clk Div2** source instead of **sys PLL clk** source. Divide the core clocking frequency by two to avoid exceeding the maximum EMIB to core frequency specification.

Figure 100. TX PMA Interface Options



5.2.3. Setting RX Datapath Options

Specify options for the following on the F-Tile PMA/FEC Direct PHY Intel FPGA IP parameter editor **RX Datapath Options** tab:

- RX FGT CDR
- RX datapath FIFO modes

The design specifies the following **RX Datapath Options**:

Table 96. RX FGT CDR Options

Parameter	Parameter Value
RX FGT CDR reference clock frequency	Select **156.25MHz**. The **RX FGT CDR reference clock frequency** must match the reference clock frequency that the F-Tile Reference and System PLL Clocks Intel FPGA IP specifies. To connect the `out_refclk_fgt_0` to this IP, refer to Connecting the F-tile PMA/FEC Direct PHY Design IP on page 212

Figure 101. RX FGT CDR Options

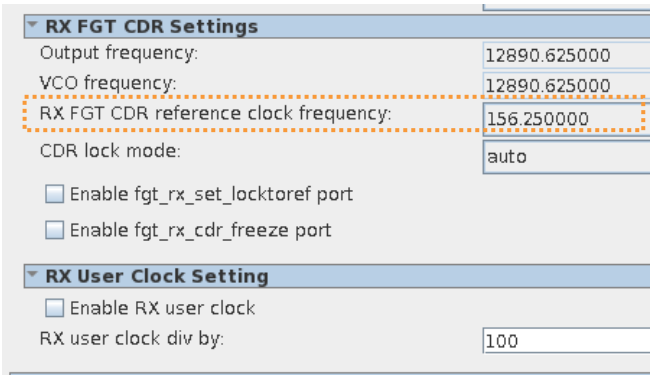
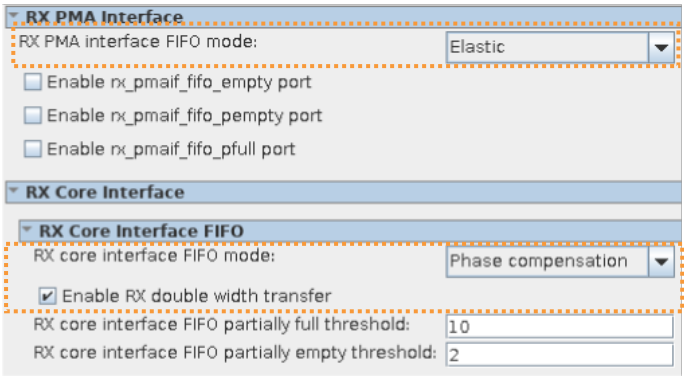


Table 97. RX PMA Interface Options

Parameter	Parameter Value
RX PMA interface FIFO mode	**Elastic**
RX core Interface FIFO Mode	**Phase Compensation**
Enable RX double width transfer	On *Note:* When you enable this option, you must drive the `tx_clkout` source with **Sys PLL Clk Div2** source instead of **sys PLL clk** source. Divide the core clocking frequency by two in this way to avoid exceeding the maximum EMIB to core frequency specification.

The RX F-tile Interface FIFO mode is always designed to be in **Register** mode for PMA direct mode, and you cannot select a different option for this IP.

Figure 102. RX PMA Interface Options



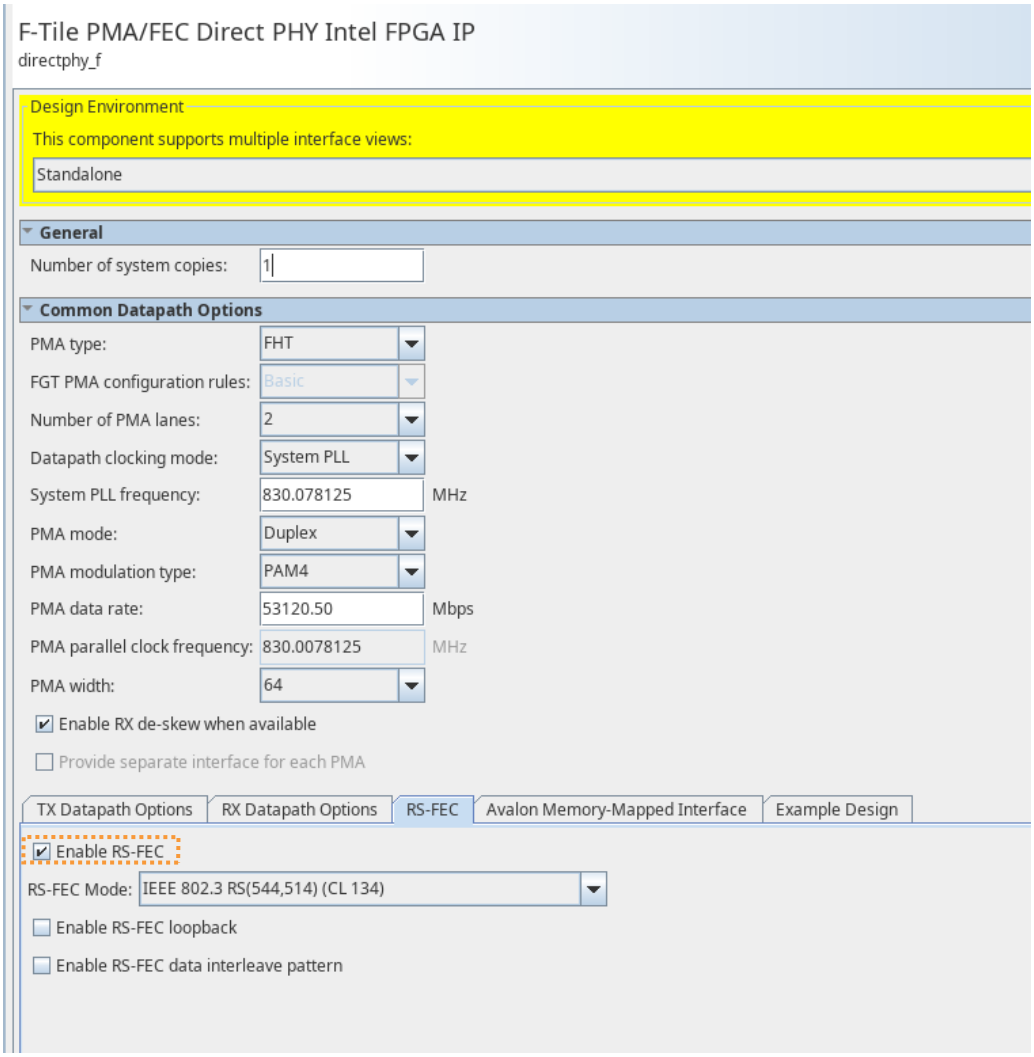
5.3. Implementing a RS-FEC Direct Design in the F-Tile PMA/FEC Direct PHY Intel FPGA IP

To instantiate a RS-FEC direct design, follow the steps mentioned in section Instantiating the F-Tile Reference and System PLL Clocks Intel FPGA IP on page 209. In addition to the PMA data path parameterization, a RS-FEC direct design allows you to enable the RS-FEC mode for forward error correction in a NRZ or PAM4 design configuration.

The RS-FEC option in F-Tile PMA/FEC Direct PHY Intel FPGA IP supports the RS-FEC modes specified in section FEC Architecture on page 16. Also refer to F-Tile Supported FEC Modes and Compliance Specifications for a comprehensive list of RS-FEC modes in the F-Tile PMA/FEC Direct PHY Intel FPGA IP. Additionally, refer to section FEC Placement Rules on page 47 for rules to follow when configuring a RS-FEC direct design.

In the RS-FEC tab of the F-Tile PMA/FEC Direct PHY Intel FPGA IP, you select the **Enable RS-FEC** to configure a design with FEC as shown in the following figure.

Figure 103. Enabling RS-FEC Mode in the IP Parameter Editor



Based on the RS-FEC mode and data rate for your design, you align the **PMA parallel clock frequency** and choose the **System PLL frequency** accordingly. Ensure that the **System PLL frequency** you choose in the F-Tile PMA/FEC Direct PHY Intel FPGA IP aligns with the F-Tile Reference and System PLL Clocks Intel FPGA IP reference clock frequency.

There are additional options that you can enable for your RS-FEC design. To enable the loopback select the **Enable RS-FEC loopback** option. You can also enable the **Enable RS-FEC data interleave pattern** option. When you enable this option, the RS-FEC lanes are bit-interleaved on each physical lane by 64/80 bits. The default value is **Off**.

When you enable the RS-FEC feature in your design, the TX and RX deskew logic is enabled. Refer to section Deskew Logic on page 20 for more information about the deskew logic in the TX and RX datapath.

In a FEC direct design, during reset sequencing, after `tx_am_gen_start` is asserted, start sending the alignment markers and assert `tx_am_gen_2x_ack` after two alignment markers are sent. The `tx_am_gen_start` goes high as part of the reset sequence, before `tx_ready` is asserted. In addition, in the FEC direct mode, you can pace the TX data valid signal with the `tx_cadence` signal.

For example, in the 100G FEC direct design, the alignment marker (AM) cycle is 81920 clock cycles and the AM pulse width is 5 clock cycles wide. In addition, the TX data is unscrambled. If a FEC direct design does not lock or align, the RX data is zero.

The FEC hard IP core has a parameter that determines the length of the alignment marker period, and is set to 10. The FEC hard IP core not only checks for the AM period, but also for the AM pulse width.

Table 98. FEC Mode AM Pulse Width Duration

FEC Mode	AM Pulse Width (Number of Cycles at IP Interface)
25G	4
50G	2
100G	5
128GFC, 200G, 400G	2
32GFC	1
64GFC	2

When FEC is configured in 200G or 400G variant modes, you should not scramble or descramble the data as this is done by the RS-FEC hard IP. In all other FEC configurations, such as 25G, 50G or 100G, you have to scramble the input data and you have to descramble output data.

5.4. Instantiating the F-Tile Reference and System PLL Clocks Intel FPGA IP

The F-Tile Reference and System PLL Clocks Intel FPGA IP is required for F-tile PMA/FEC Direct PHY designs. You must instantiate and connect this IP for simulation and compilation.

This design requires the following for the F-Tile Reference and System PLL Clocks Intel FPGA IP:

- System PLL mode and reference clock source for one system PLL that clocks the datapath.
- Reference clock source for FGT PMA. Share or separate the reference clock source for system PLL and FGT PMA. This example shares the reference clock.

To specify the system PLL mode and reference clock source for one system PLL:

1. In the IP Catalog search field, type `f-tile Reference`, and double-click the **F-Tile Reference and System PLL Clocks Intel FPGA IP** under **Transceiver PHY**.
2. On the **System PLL #0** tab, specify the following:

Table 99. System PLL #0 Options

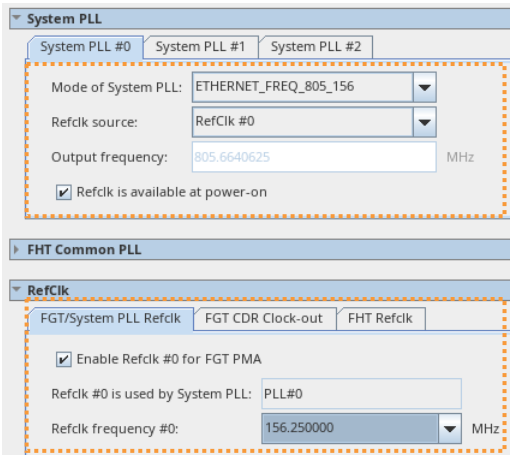
Parameter	Parameter Value
Mode of system PLL	Select **ETHERNET_FREQ_830_156**. This configures the system PLL input frequency to 156.25 MHz and output frequency to 830.078125 MHz (must match system PLL frequency in F-Tile PMA/FEC Direct PHY Intel FPGA IP).
Refclk source	**RefClk #0**. Selects the reference clock source for system PLL.
RefClk #0 for FGT PMA	On

3. Under **RefClk**, specify the following options:

Table 100. RefClk Options

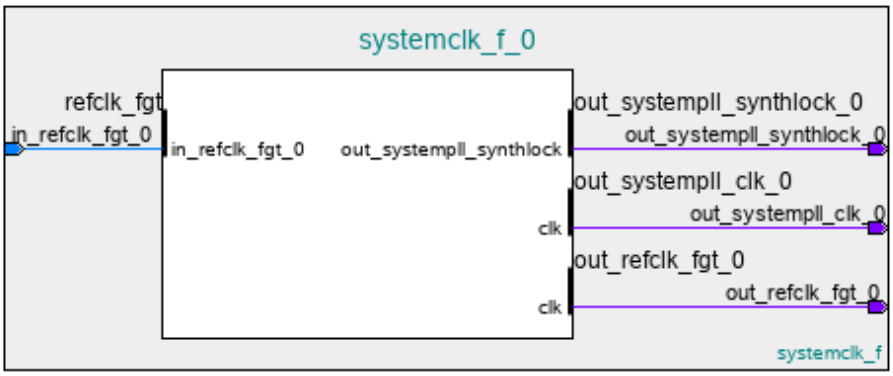
Parameter	Parameter Value
Enable RefClk #0 for FGT PMA	On. This reference clock is shared between system PLL #0 and FGT PMA
RefClk source	**RefClk #0**. 156.25 MHz (same as reference clock frequency for F-Tile PMA/FEC Direct PHY Intel FPGA IP).

Figure 104. System PLL and RefClk Options



The following figures shows the block symbol and available ports for F-Tile Reference and System PLL Clocks Intel FPGA IP in this example

Figure 105. Example F-Tile Reference and System PLL Clocks Intel FPGA IP Ports

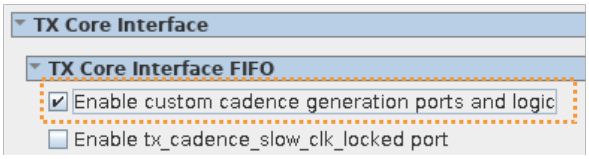


5.5. Enabling Custom Cadence Generation Ports and Logic

This F-tile PMA/FEC Direct PHY design uses System PLL clocking mode to clock the digital datapath of the FGT PMA lane. Because the system PLL frequency (830.078125MHz) is greater than the PMA clock frequency (805.6640625MHz), you must enable custom cadence generation logic ports, and enable the logic option in the IP parameter editor.

- You must use `tx_cadence` port output to assert and de-assert the TX PMA Interface data valid bit (one of the bits in TX parallel data). Refer to Parallel Data Mapping Information on page 124.
- You must connect `tx_cadence_fast_clk` to `tx_clkout/tx_clkout2` with clock source **System PLL Clock** / 2 (415.0390625MHz).
- You must connect `tx_cadence_slow_clk` to `tx_clkout/tx_clkout2` with clock source **Word clock** or **Bond clock** / 2 (402.83203125 MHz)

Figure 106. Enabling Custom Cadence Generation Ports and Logic



Rate Match FIFO Requirement

The following guidelines apply to the elastic FIFO requirement between user FPGA core logic and the F-Tile PMA/FEC Direct PHY Intel FPGA IP:

- If the user FPGA core logic is running at same frequency as system PLL frequency/2 (that is, 415.0390625MHz), then there is no elastic FIFO requirement between the user FPGA core logic and the F-Tile PMA/FEC Direct PHY Intel FPGA IP.
- If the user FPGA core logic is running at PMA clock frequency/2 (that is, 402.83203125 MHz), this requires elastic FIFO between the user FPGA core logic and the F-tile core interface FIFO to transfer from PMA clock frequency domain to system PLL clock frequency domain and must be implemented by the user.

Related Information

- Custom Cadence Generation Ports and Logic on page 145
- TX PMA interface Parameters on page 95

5.6. Connecting the F-tile PMA/FEC Direct PHY Design IP

After generating the RTL and supporting files for the F-Tile PMA/FEC Direct PHY Intel FPGA IP and F-Tile Reference and System PLL Clocks Intel FPGA IP, you connect the two IP together in the top level file (`top.v`) based on the connections in F-tile PMA/FEC Direct PHY Design IP Connections. Verify the top-level connection before running the **Design Analysis** Compiler stage.

Table 101. F-tile PMA/FEC Direct PHY Design IP Port Connections

F-Tile Reference and System PLL Clocks Intel FPGA IP Ports	F-Tile PMA/FEC Direct PHY Intel FPGA IP Ports
`out_refclk_fgt_0`	• `tx_pll_refclk_link`[44] • `rx_cdr_refclk_link`
`out_systempll_clk_0`	`System_pll_clk_link`

5.7. Simulating the F-Tile PMA/FEC Direct PHY Design

Simulation of the F-tile PMA/FEC Direct PHY design requires running the Intel Quartus Prime Compiler's **Analysis & Elaboration** and **Support-Logic Generation** stages to elaborate the component IP in your design. Next, you generate the simulator setup scripts for the Synopsys VCS simulator or ModelSim simulator. You can modify and use the setup scripts to elaborate and simulate your design and testbench.

The simulation flow for F-tile differs from other serial interface tiles. The difference in the F-Tile PMA/FEC Direct PHY Intel FPGA IP simulation flow is that you cannot simulate individual IP files.

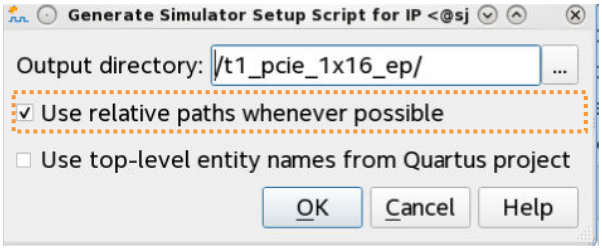
To simulate the F-tile PMA/FEC Direct PHY design using VCS:

1. Click **Processing ➤ Start ➤ Start Analysis & Elaboration**. Review and correct any error messages.
2. On the Compilation Dashboard, click **Support-Logic Generation**. Review and correct any error messages.

   Note: A `<top_level>_auto_tiles.sv` file is auto-generated in the `support_logic` folder. For example, when the top-level entity is top, the generated file is `top_auto_tiles.sv`. You have to instantiate this in your test bench.
3. Click **Tools ➤ Generate Simulator Setup Script for IP** and retain the default options. Ensure that you do not turn on the **Use top-level entity names from Quartus project** option.

(44) Ports ending in "_link" must connect to the F-Tile Reference and System PLL Clocks Intel FPGA IP. These ports cannot be simulated

Figure 107. Generate Simulator Setup Script for IP Dialog Box



4. Click **OK**. The `vcs_setup.sh` file generates in the `synopsys/vcs` directory and `vcs_files.tcl` generates in the `common` directory.
5. Navigate to the `/synopsys/vcs` directory.
6. Review the instructions in `vcs_setup.sh` to create the `vcs_sim.sh` file for your simulation. The following shows an example `vcs_sim.sh`. Your `vcs_sim.sh` content varies by your design characteristics.

```
source ./vcs_setup.sh \
TOP_LEVEL_NAME=top_tb \
QSYS_SIMDIR=../.. \
USER_DEFINED_ELAB_OPTIONS="\"-full64 +v2k -hsopt=gates \
+systemverilogext+.sv -sverilog -lca +lint=TFIPC-L +lint=PCWM -wreal res_def
\
 -xlrm coerce_nettype -timescale=1ns/1ps +vcs+vcdpluson +vpddrivers \
 +define+TIMESCALE_EN +define+INTC_FUNCTIONAL +define+RTLSIM \
+define+SSM_SEQUENCE \
+define+IP7581SERDES_UXS2T1R1PGD_PIPE_SPEC_FORCE \
+define+IP7581SERDES_UXS2T1R1PGD_PIPE_SIMULATION \
+define+IP7581SERDES_UXS2T1R1PGD_PIPE_FAST_SIM +error+1000 \
+define+__SRC_TEST__ -debug_access+r+driver+f -debug_region+encrypt \
 +rad -f ./filelist.f  \" " \
USER_DEFINED_ELAB_OPTIONS_APPEND="\" -l vcs.log\" "
./simv +lic+wait -l simulation.log +fsdb+mda=on +fsdbfile+novas.fsdb
```

Note: In order to provide a reduction in real-time simulation duration, you can use a *Fast Sim* model. This model is enabled via a macro in the simulation run scripts. The syntax to enable the *Fast Sim* model is as follows:

```
+define+IP7581SERDES_UX_SIMSPEED
```

7. Create your test bench. When calling the top level module in the test bench, the module name must be the same as instance name. For example:

```
top top (
ports passing
);
```

A mismatch between the module and instance names causes errors during compilation. For example, the following produces errors during compilation:

```
top top_instant0 (
ports passing
);
```

Also be sure to add this instantiation in your test bench:

```
top_auto_tiles <any names> ();
```

For example, *<any names>* can be `top_auto_tiles_inst1`.

8. Create the `filelist.f` file. `vcs_setup.sh` contains the instructions for `filelist.f`. `filelist.f` contains all your testbench files and top-level. Note that the file location is relative to the `vcs_sim.sh` command. The following shows one example `filelist.f`. Your `filelist.f` content varies, depending on your design characteristics.

```
# # DESIGN FILE LIST & OPTIONS TEMPLATE - BEGIN
# #
# # Compile all design files and testbench files, including the top level.
# # (These are all the files required for simulation other than the files
# # compiled by the Quartus-generated IP simulation script)
# #
# +systemverilogext+.sv
# <design and testbench files, compile-time options, elaboration options>
# #
# # DESIGN FILE LIST & OPTIONS TEMPLATE - END

../../top.v
../../top_tb.v
```

9. Start VCS compilation and simulation.
10. Launch waveform viewer.

5.7.1. PAM4 Encoding Schemes in Simulation

When simulating the FGT or FHT transceivers in PAM4 mode, the following tables represent the encoding schemes for the positive (_p) and negative (_n) serial lines:

Table 102. FGT PAM4 Encoding Scheme

Non-Gray Input MSB, LSB	Gray Input MSB, LSB	Simulation Output for P, N
11	10	1, Z
10	11	1, 0
01	01	0, Z
00	00	0, 1

Table 103. FHT PAM4 Encoding Scheme

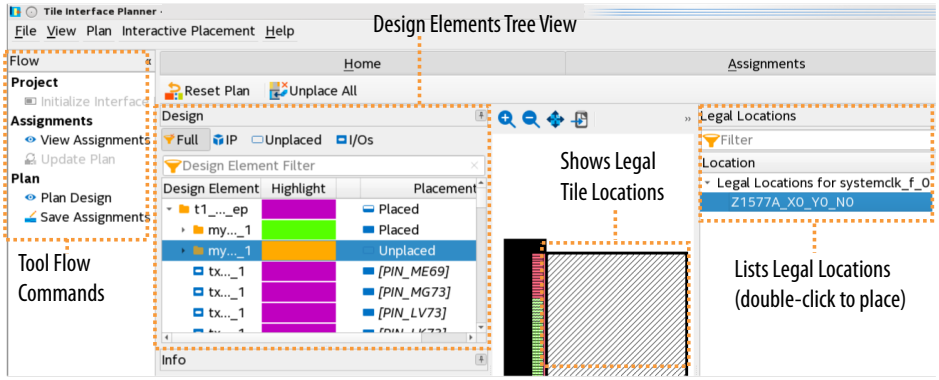
Input MSB, LSB	Simulation Output for P, N
11	11
10	10
01	01
00	00

5.8. F-tile Interface Planning

The decomposable Intel Agilex 7 F-tile architecture prompts a new tile planning step for PHY layer implementation. This step allows you to place component IP in specific device tiles to reflect your board or system level constraints. The Intel Quartus Prime Tile Interface Planner simplifies placement of component IP in legal tile locations.

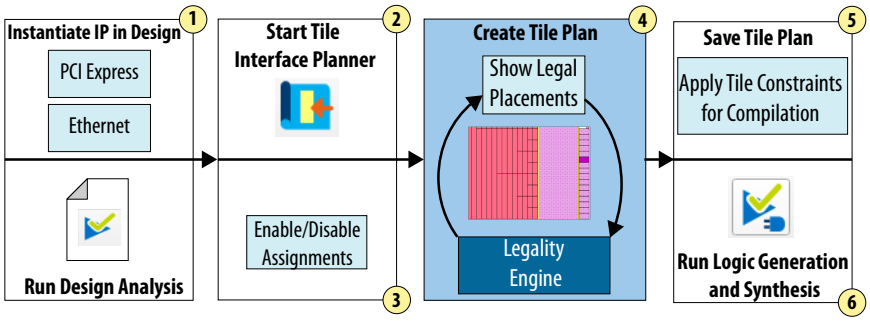
Tile Interface Planner displays your design's component IP in a hierarchical view, next to a visual display of the device tile fractures. You locate legal tile locations, place the IP, and save the placement constraints for downstream Compiler stages. The legality engine verifies placement in real-time to ensure correlation in final implementation.

Figure 108. Tile Interface Planner



Tile Interface Planner guides you through the tile planning steps:

Figure 109. Tile Interface Planner Tool Flow



Refer to *Tile Interface Planning* in the *Intel Quartus Prime Pro Edition User Guide: Design Constraints* for Tile Interface Planner use information.

Related Information

Intel Quartus Prime Pro Edition User Guide: Design Constraints

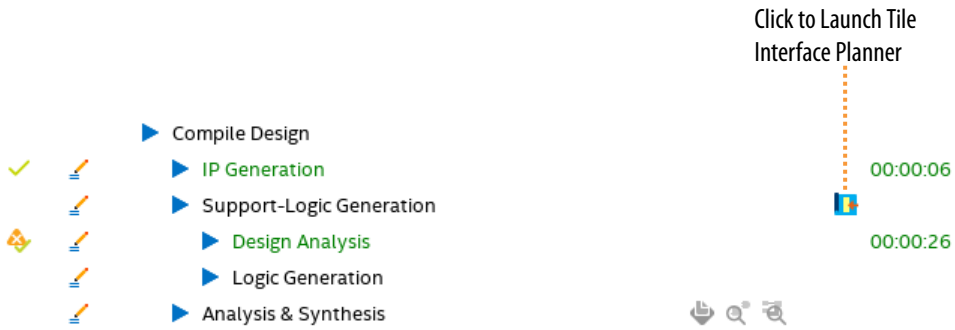
5.8.1. F-tile Interface Planner Usage Example

The design used includes two 25.78125 Gbps NRZ PMA Direct FGT PMA lanes, with a throughput of 51.5625 Gbps, and with the system PLL datapath clocking mode.

The example illustrates the steps you need to follow to use the Tile Interface Planner tool in the Intel Quartus Prime software.

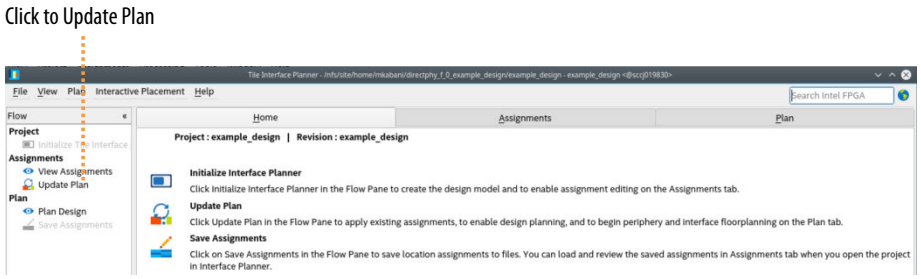
1. Run the **Design Analysis** substep under **Support-Logic Generation** in the compilation flow window of Intel Quartus Prime software.
2. Click the Tile Interface Planner tool icon on the right side of the compilation flow window to launch the tool as shown in the following figure.

Figure 110. Launching the Tile Interface Planner



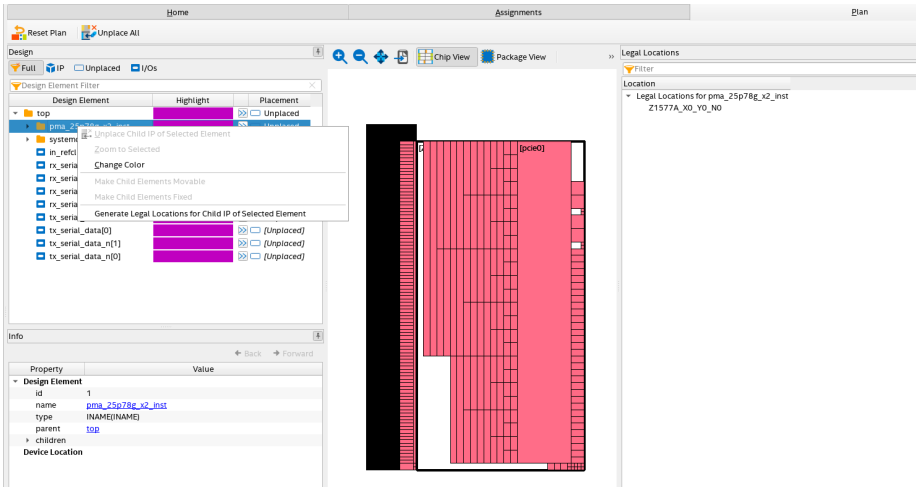
3. When the tool successfully launches, click on **Update Plan** under the **Flow** pane on the left side to load any saved plans and begin tile interface planning as shown in the following figure.

Figure 111. Update Plan in the Tile Interface Planner



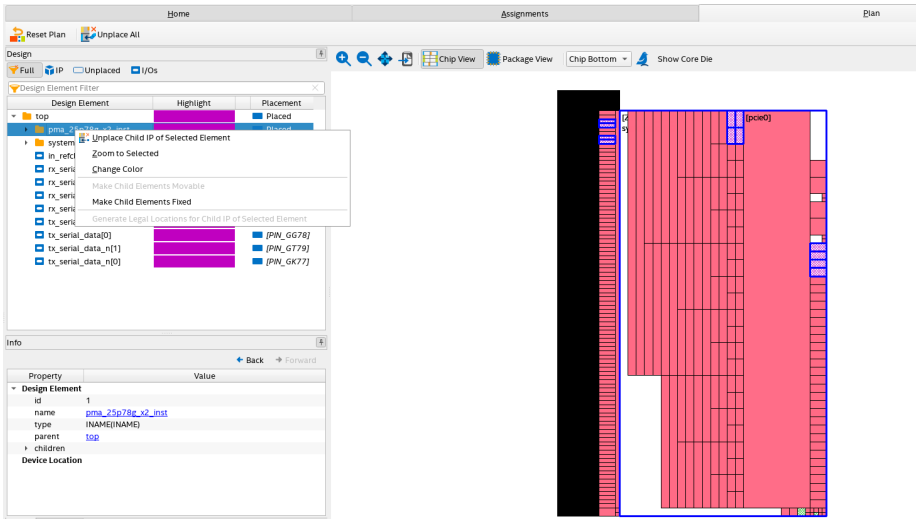
4. Navigate to the **Plan** tab to visualize the design elements and the tile floorplan. Right click on any of the design elements to see the available legal locations in the right pane for that element, and double click on one of the locations to place the IP element as shown in the following figure.

Figure 112. Place Elements in the Tile Interface Planner



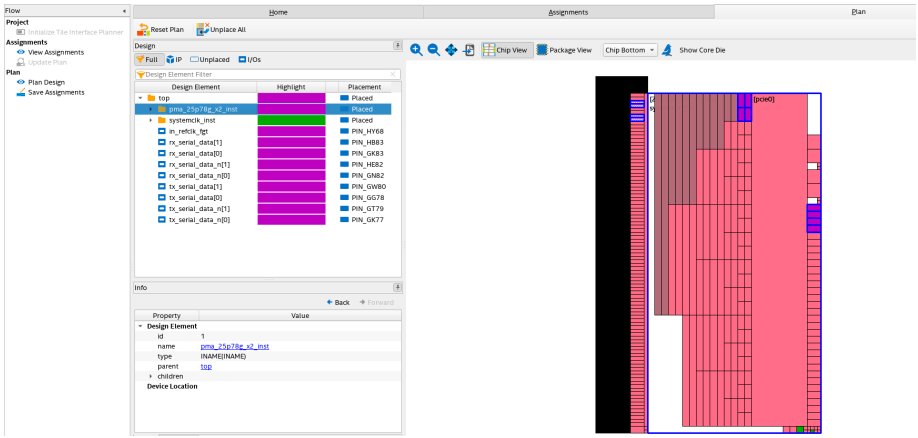
5. Right click on any of the design elements to make them fixed and to save the placement as shown in the following figure.

Figure 113. Save Placement in the Tile Interface Planner



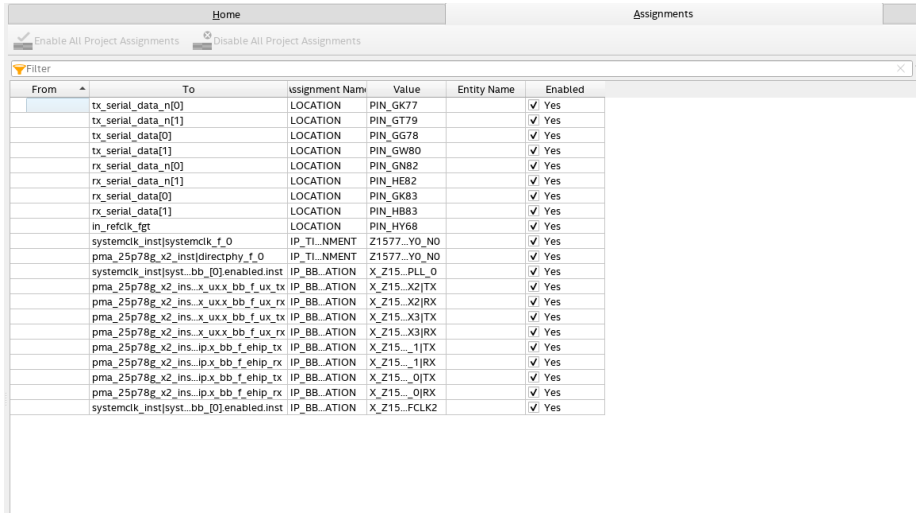
6. Click on **Save Assignments** in the **Flow** pane on the left, to save the placements as `.qsf` assignments as shown in the following figure.

Figure 114. Save Assignments in the Tile Interface Planner



7. Navigate to the **Assignments** tab, to see the saved `.qsf` assignments for the design as shown in the following figure.

Figure 115. View Assignments in the Tile Interface Planner



From	To	ssignment Nam	Value	Entity Name	Enabled
	tx_serial_data_n[0]	LOCATION	PIN_GK77		Yes
	tx_serial_data_n[1]	LOCATION	PIN_GT79		Yes
	tx_serial_data[0]	LOCATION	PIN_GG78		Yes
	tx_serial_data[1]	LOCATION	PIN_GW80		Yes
	rx_serial_data_n[0]	LOCATION	PIN_GN82		Yes
	rx_serial_data_n[1]	LOCATION	PIN_HE82		Yes
	rx_serial_data[0]	LOCATION	PIN_GK83		Yes
	rx_serial_data[1]	LOCATION	PIN_HB83		Yes
	in_refclk_fgt	LOCATION	PIN_HY68		Yes
	systemclk_inst|systemclk_f_0	IP_TI...NMENT	Z1577...Y0_N0		Yes
	pma_25p78g_x2_inst|directphy_f_0	IP_TI...NMENT	Z1577...Y0_N0		Yes
	systemclk_inst|syst...bb_[0].enabled.inst	IP_BB...ATION	X_Z15...PLL_0		Yes
	pma_25p78g_x2_ins...x_ux_bb_f_ux_tx	IP_BB...ATION	X_Z15...X2|TX		Yes
	pma_25p78g_x2_ins...x_ux_bb_f_ux_rx	IP_BB...ATION	X_Z15...X2|RX		Yes
	pma_25p78g_x2_ins...x_ux_bb_f_ux_tx	IP_BB...ATION	X_Z15...X3|TX		Yes
	pma_25p78g_x2_ins...x_ux_bb_f_ux_rx	IP_BB...ATION	X_Z15...X3|RX		Yes
	pma_25p78g_x2_ins...ip.x_bb_f_ehip_tx	IP_BB...ATION	X_Z15..._1|TX		Yes
	pma_25p78g_x2_ins...ip.x_bb_f_ehip_rx	IP_BB...ATION	X_Z15..._1|RX		Yes
	pma_25p78g_x2_ins...ip.x_bb_f_ehip_tx	IP_BB...ATION	X_Z15..._0|TX		Yes
	pma_25p78g_x2_ins...ip.x_bb_f_ehip_rx	IP_BB...ATION	X_Z15..._0|RX		Yes
	systemclk_inst|syst...bb_[0].enabled.inst	IP_BB...ATION	X_Z15...FCLK2		Yes

6. Supported Tools

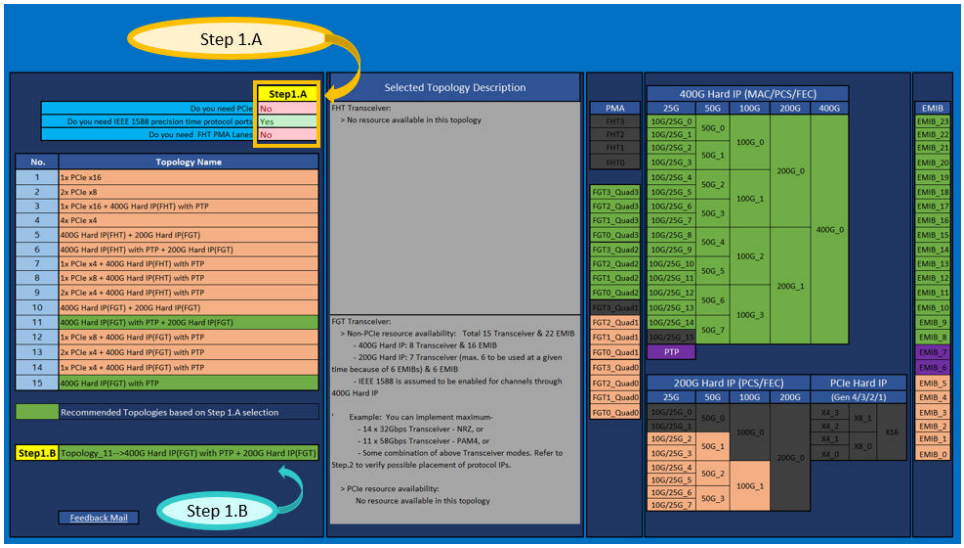
6.1. F-Tile Channel Placement Tool

F-tile supports datacenters, 5G networks, Smart Grid and other market segments. Ethernet, CPRI and OTN are the backbone of these emerging and traditional technologies. The *F-Tile Channel Placement Tool*, in conjunction with the *Device Family Pin Connection Guidelines*, allows you to swiftly plan protocol placements in the product prior to reading comprehensive documentation and implementing designs in Intel Quartus Prime software.

The Excel-based *F-Tile Channel Placement Tool*, supplemented with the following tabs is available for download at F-Tile Channel Placement Tool:

- **Instructions**
- **PMA Hard IP Mapping (Reference)**
- **Step1_Topology Selection**
- **Step2 Hard IP Transceiver Placement**
- **Revision**

Figure 116. F-Tile Channel Placement Tool



Related Information

- F-Tile Channel Placement Tool
- Intel Agilex 7 Device Family Pin Connection Guidelines

6.2. F-Tile PMA and FEC Direct Port Mapping Calculator

The *F-Tile PMA and FEC Direct Port Mapping Calculator* helps you to calculate the following:

- TX and RX parallel data bus width.
- Location of TX and RX parallel data bits.
- Control bits such as data valid.
- Write enable for TX core interface FIFO
- RX deskew and data valid for RX Core FIFO in elastic mode.
- FEC bits such as sync head based on the input configuration.

Note: More information about the FEC bits such as sync head based on the input configuration can be found in Bit Mapping for PMA and FEC Mode PHY TX and RX Datapath on page 121 .

The Excel-based *F-Tile PMA and FEC Direct Parallel Data Mapping Calculator*, is available for download at F-Tile PMA and FEC Direct Port Mapping Calculator

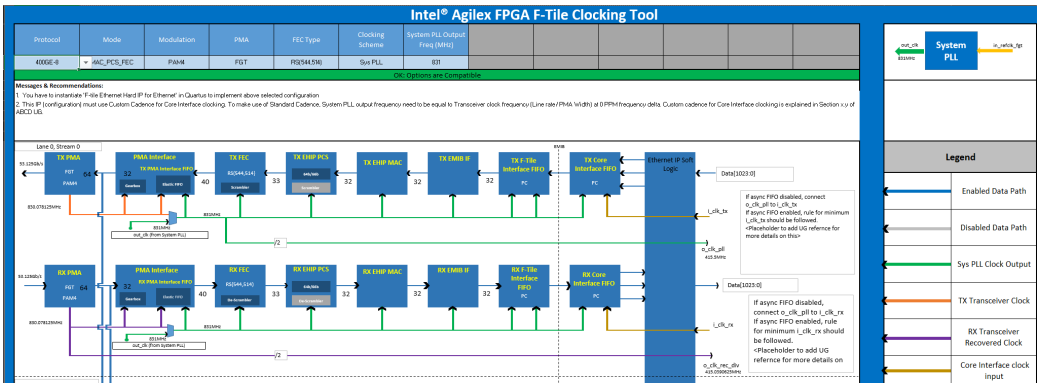
6.3. F-Tile Clocking and Datapath Tool

The *F-Tile Clocking and Datapath Tool* helps you to configure the F-Tile PMA/FEC Direct PHY Intel FPGA IP and displays the following:

- TX and RX datapath of one configuration at a time.
- Selection of parameters that can affect clocking.
- Single vs multiple streams based on chosen configuration.
- Parameters such as FEC type, clocking scheme, PMA width, and core and interface FIFO.

The excel based *F-Tile Clocking and Datapath Tool* is available for download at F-Tile Clocking and Datapath Tool. A screenshot of the tool is shown in the following figure.

Figure 117. F-Tile Clocking and Datapath Tool



Note: The number of streams and PMA lanes change according to the configuration chosen in the tool.

6.4. F-Tile 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 F-Tile FGT TX PMA offers pre-emphasis and linear equalization to address this problem and improve the high-speed signal quality.

You can use the *F-Tile TX Equalizer Tool* to estimate the pre-emphasis and output swing. The tool displays the FGT PMA transmitter output swing when you change transmitter's:

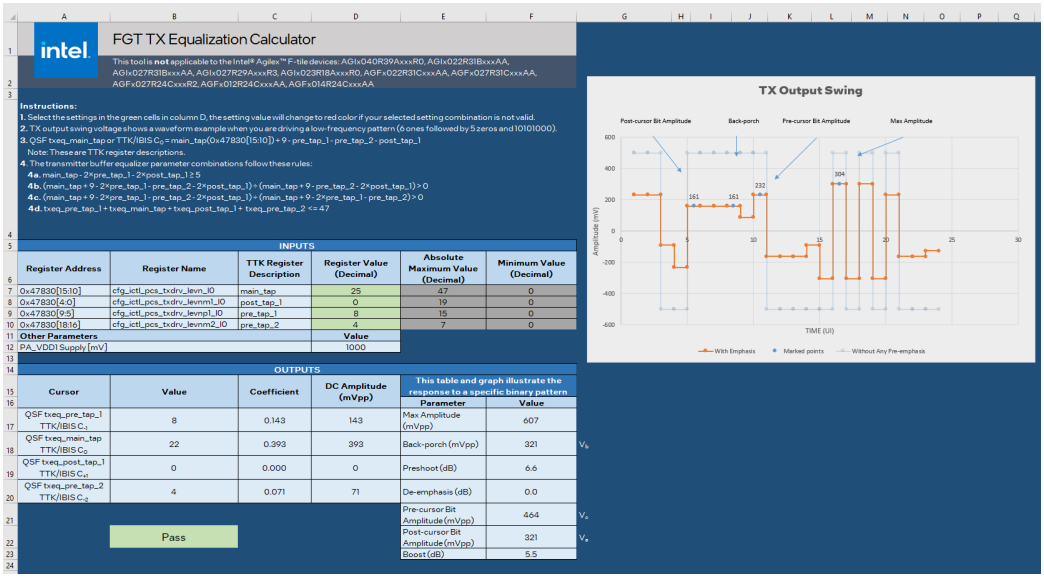
- Main_tap
- Post_tap_1
- Pre_tap_1
- Pre_tap_2

The Excel-based *F-Tile TX Equalizer Tool*, is available for download:

- For Intel Agilex 7 F-tile ES devices, excluding OPNs mentioned in *Appendix A.1*: F-Tile TX Equalizer Tool
- For Intel Agilex 7 F-tile devices with OPNs mentioned in *Appendix A.1*: F-Tile TX Equalizer Tool

A screenshot of the tool is shown below:

Figure 118. F-Tile TX Equalizer Tool



7. Debugging F-Tile Transceiver Links

The Transceiver Toolkit helps you optimize high-speed serial links in your board design by providing real-time control, monitoring, and debugging of the F-tile transceiver links running on your board.

The F-Tile Transceiver Toolkit allows you to:

- Control the transmitter or receiver channels to optimize transceiver settings and hardware features.
- Test bit-error rate (BER) while running multiple links at the target data rate.
- Control internal pattern generators and checkers.
- Run autosweep tests to identify the best PMA settings for each link.
- View the receiver horizontal and vertical eye margin during testing.
- Test multiple devices across multiple boards simultaneously.

Note: The F-Tile Transceiver Toolkit runs from the System Console framework.

7.1. F-Tile Transceiver Toolkit GUI

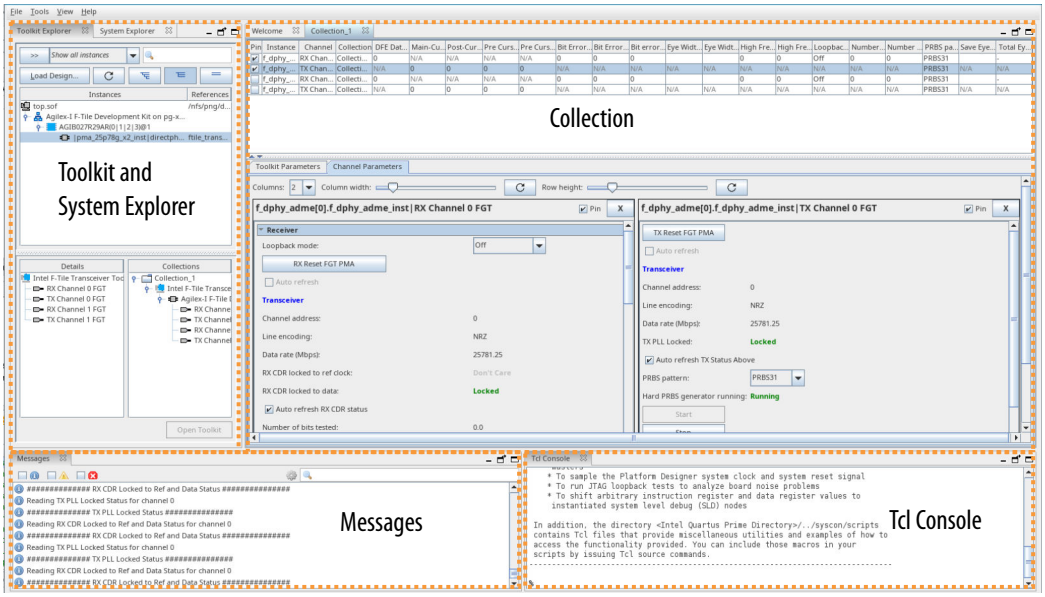
The F-Tile transceiver toolkit GUI consists of a main window with multiple panes that allow you to interact with the design currently running on the host computer.

The F-Tile transceiver toolkit displays the following in the GUI window:

- **Collection**: Allows you to configure and control transceiver channels and links, and adjust programmable analog settings to improve the signal integrity of the link.
- **Toolkit Explorer**: Displays all available toolkits and launches tools that use the System Console framework.
- **System Explorer**: Displays a list of interactive instances in your design, including board connections, devices, designs, servers, and scripts.
- **Tcl Console**: Allows you to interact with your design using Tcl scripts, for example, sourcing scripts, writing procedures, and using System Console API.
- **Messages**: Displays status, warning, error messages related to connections and debug actions.

The following figure shows the F-Tile transceiver toolkit GUI.

Figure 119. F-Tile Transceiver Toolkit GUI



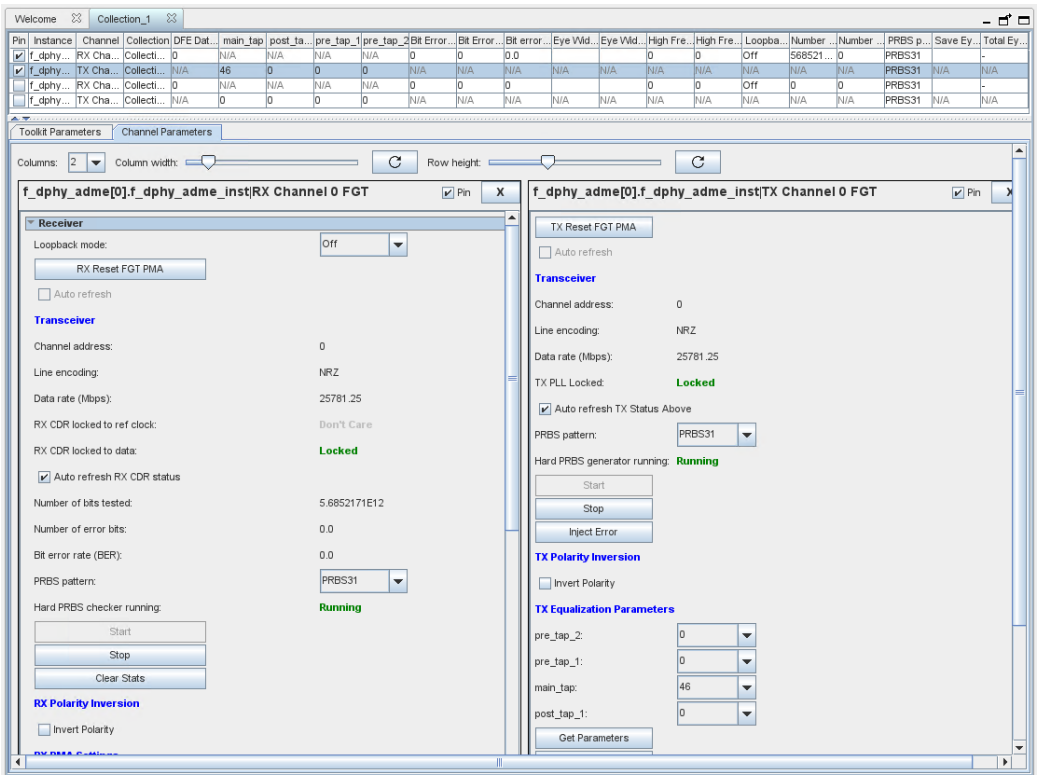
7.1.1. Collection View

The **Collection** view consists of three main panes:

- **Status Table**: You can view and configure channels from different instances in a single view. You can select the channels you want to configure and display in the **Channel Parameters**. By choosing the desired channels, right-clicking, and exploring the **Actions** sub-menu, you can perform bulk actions across multiple channels. You can customize the status table by selecting the parameters you want to show in the table. Right click at the top of the table and select **Edit Columns**. A **Select column** headers window opens up which lists all the parameters of a channel. Select the parameters you want and click **OK**.
- **Toolkit Parameters**: You can view and configure the Autosweep and Eye Viewer settings in this pane. The auto refresh period settings are going to be enabled in future Intel Quartus Prime software versions.
- **Channel Parameters**: You can control, monitor channel settings and status, and measure on-die eye margin in this pane. You can start and stop bit error rate tests by clicking the **Start** and **Stop** buttons. When you select more than one channel, you can view all channels in one tab by adjusting the number of columns. You can also control the width of the column and height of the row to fit all the channels in one tab.

The following figure shows the **Collection** view of the F-Tile transceiver toolkit GUI.

Figure 120. Collection View Tab of the F-Tile Transceiver Toolkit GUI



7.2. F-Tile Transceiver Debugging Flow Walkthrough

These steps describe the high-level process of debugging F-Tile transceivers using the transceiver toolkit.

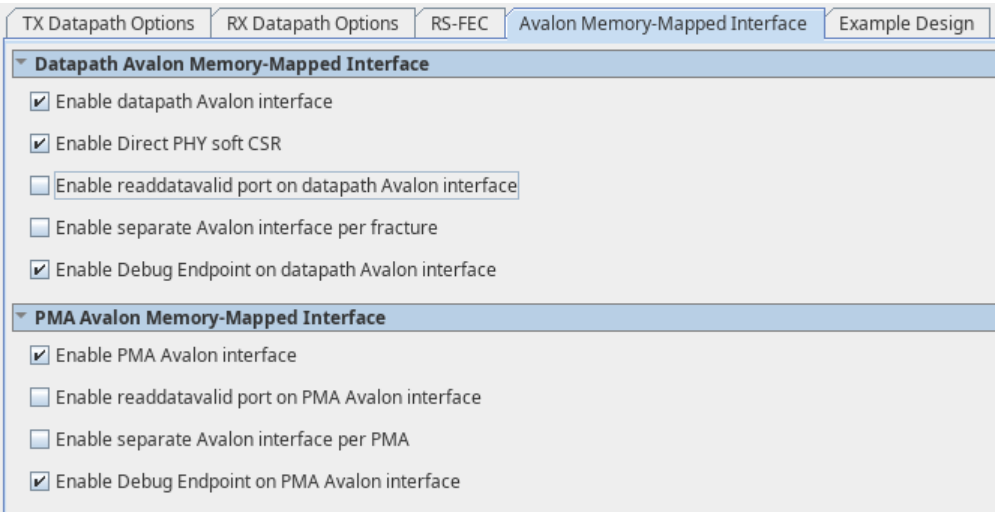
1. Modifying the design to enable F-Tile transceiver debug.
2. Programming the design into an Intel FPGA.
3. Loading the design to the F-Tile transceiver toolkit.
4. Creating transceiver links.
5. Running BER tests.
6. Running link optimization tests.

7.2.1. Modifying the Design to Enable F-Tile Transceiver Debug

To enable debugging capabilities, you must enable the Avalon memory-mapped interface parameters in the F-Tile PMA/FEC Direct PHY Intel FPGA IP.
You can either activate these settings when you first instantiate the IP or modify the instances after preliminary compilation. Follow these steps to enable the settings:

1. In the **IP Components** tab of the **Project Navigator**, right click the IP instance, and select **Edit in Parameter Editor**.
2. Enable the datapath and PMA Avalon interface, Direct PHY soft CSR, and debug endpoint options under the **Avalon Memory-Mapped Interface** tab as shown in the following figure.

Figure 121. Parameters to Enable Transceiver Toolkit in F-Tile PMA/FEC Direct PHY Intel FPGA IP



3. Connect the reference signals that the debugging logic requires, if applicable. The debug endpoint requires clock and reset signal connections. For details on the how to connect these signals, refer to Configuring the F-Tile PMA/FEC Direct PHY Intel FPGA IP for Hardware Testing on page 167.
4. Click **Generate HDL**. After enabling parameters for all the IP instances in the design, recompile the project.

Related Information

- Avalon Memory Mapped Interface Options on page 106
- PMA Avalon Memory Mapped Interface Signals on page 118
- Datapath Avalon Memory Mapped Interface Signals on page 120

7.2.2. Programming the Design into an Intel FPGA

After you enable the F-Tile transceiver toolkit parameters in the F-Tile PMA/FEC Direct PHY Intel FPGA IP design, you can compile, and generate the programming files. You can then program the design into an Intel FPGA.

Related Information

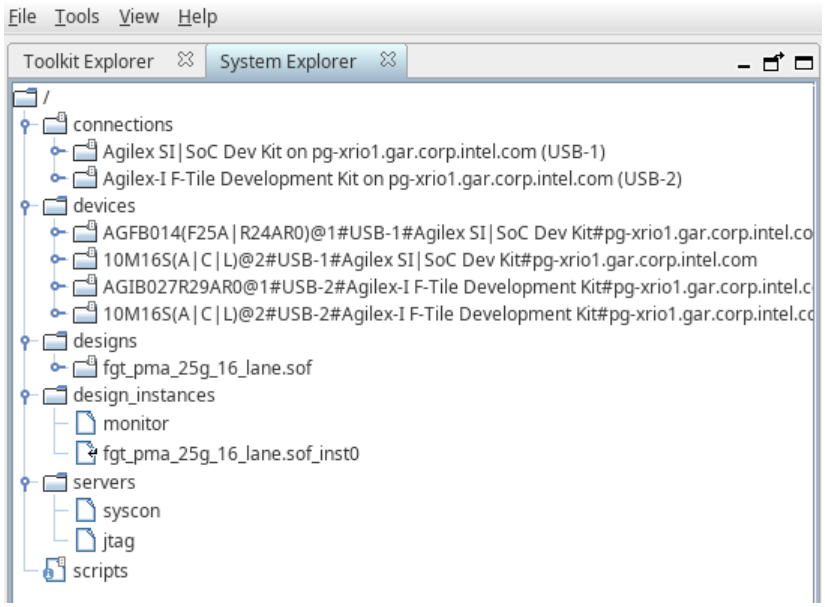
Intel Quartus Prime Pro Edition User Guide: Programmer

7.2.3. Loading the Design to the Transceiver Toolkit

To launch the toolkit, click **Tools ➤ System Debugging Tools ➤ System Debugging Toolkits**.

If the FPGA is already programmed with the project when loading, the transceiver toolkit automatically links the design to the target hardware in the toolkit. The toolkit automatically discovers links between the transmitter and receiver of the same channel. Before loading the design, ensure that you connect the hardware with the Intel FPGA. The device and JTAG connections appear in the **devices** and **connections** folders of the **System Explorer** pane as shown in the following figure.

Figure 122. System Explorer Pane



To load the design into the transceiver toolkit, follow these steps:

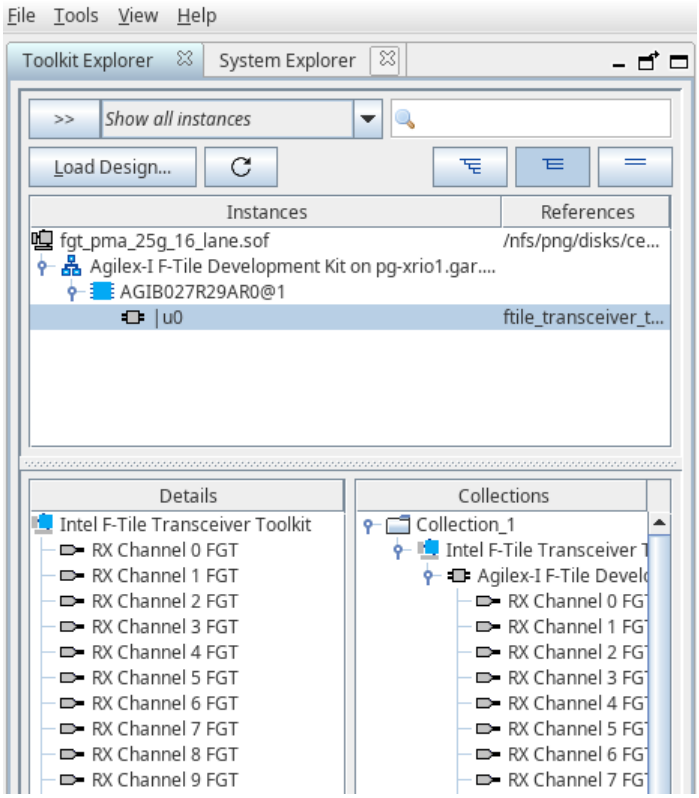
1. In System Console, click **File ➤ Load Design**.
2. Select the `.sof` programming file for the transceiver design.

After loading the project, the **designs** and **design_instances** folders in the **System Explorer** pane display information about the design, such as the design name and the blocks in the design that can communicate to the **System Console**.

7.2.4. Creating Transceiver Links

Transceiver links are identified automatically when a receiver and transmitter share the same channel. Each enabled transmitter and receiver channel on all loaded and linked devices is displayed in the **Toolkit Explorer** as shown in the following figure.

Figure 123. Toolkit Explorer



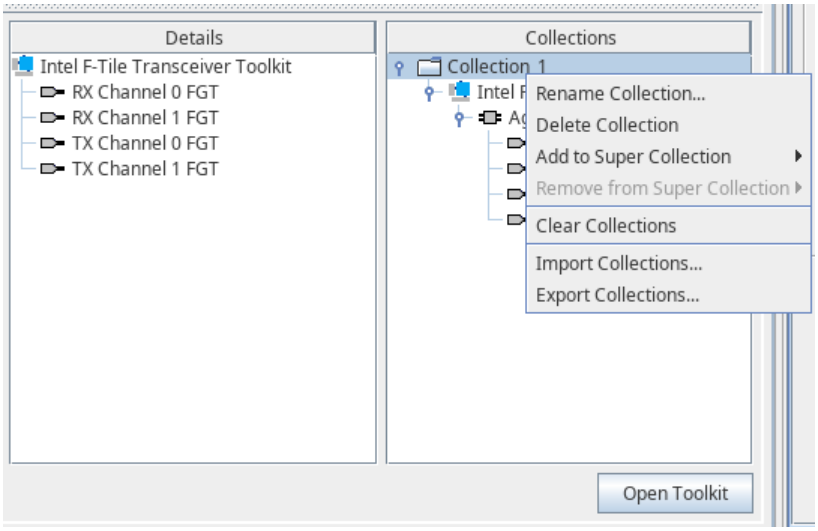
You can create a custom collection to view and configure TX and RX channels. If you want to create TX and RX paths between different physical channels, whether they are in the same device or in different devices, you must manually create new links. To have TX and RX paths between different physical channels, make sure you have an external loopback either using a loopback cable or card on the board to have physical connections between the channels.

To manually create transceiver links that have TX and RX channels in different physical channel locations, follow these steps:

1. Choose the TX and RX pair you want to link.
2. Right-click to create a collection and specify a name in the **Add to Collection** box.
3. Click **OK**. The link you create adds to the **Collections** box.
4. Click **Open Toolkit**. You can also open all the channels in one view by double-clicking the instances in the **Details** box. The name of the channels collection is automatically added.
5. Go to main **Collections** pane view where you can control and monitor the channels.

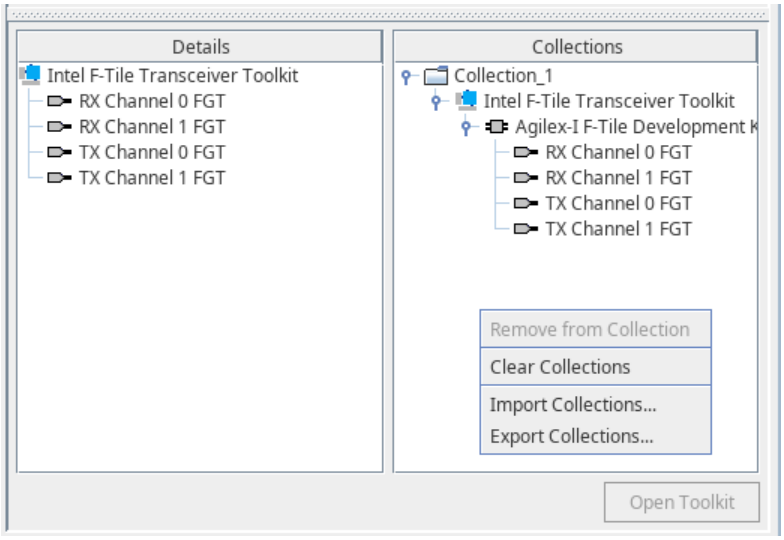
You can save the collections and load them at a later time. To save a collection, right-click on the collection and select **Export Collections** as shown in the following figure.

Figure 124. Export Collections



You can also load a previously saved collection. Right-click in the **Collections** window and select **Import Collections** as shown in the following figure.

Figure 125. Import Collections



Related Information

- Creating Collections from the Toolkit Explorer section, Intel Quartus Prime Pro Edition User Guide: Debug Tools
- System Explorer Pane section, Intel Quartus Prime Pro Edition User Guide: Debug Tools

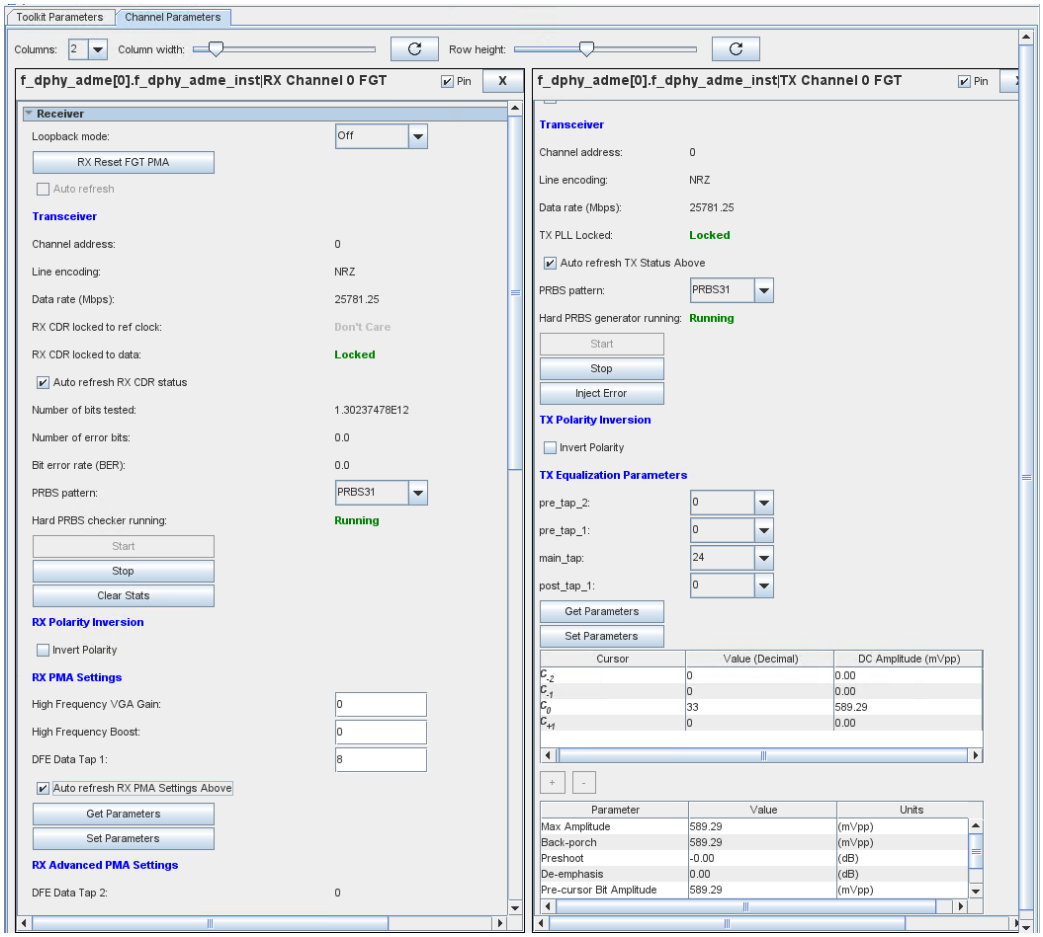
7.2.5. Running BER Tests

After you create the transceiver links for debugging, you can run BER tests from the **Channel Parameters** tab:

1. In the **Collection** status table, select the pin for the TX and RX channels you want to test.
2. In the **Channel Parameters** tab, select the PRBS pattern in the TX and RX channels.
3. Set the **TX Equalization Parameters**. Key in the values, click **Set Parameters**. To load the values, click **Get Parameters**.
4. Click **Start** in the TX Channel to start the Hard PRBS generator.
5. Click **TX Reset FGT PMA** and **RX Reset FGT PMA**. This step is only applicable to the FGT PMA.
6. Click **Start** in the RX Channel to start the Hard PRBS checker.
7. To stop the test, click **Stop** in the RX Channel and TX Channel.
8. If you want to run another BER test with a different PRBS pattern or TX equalization parameters, repeat steps 2 to 7.

The following figure shows the setup and results for an example BER test for the FGT PMA.

Figure 126. Example BER Test Setup and Results for the FGT PMA



You can set parameters, start PRBS generator, stop PRBS checker or reset across multiple channels simultaneously from the **Status Table**. Choose the desired channels, right-click, select **Edit Parameters** or the **Action** sub-menu.

Note: After changing TX or RX equalization parameters across multiple channels from the **Edit Parameters** window, you need to right click on the selected channels, select **Actions ➤ Receiver** or **Transmitter ➤ Analog ➤ Set Parameters** in order to load in the updated value.

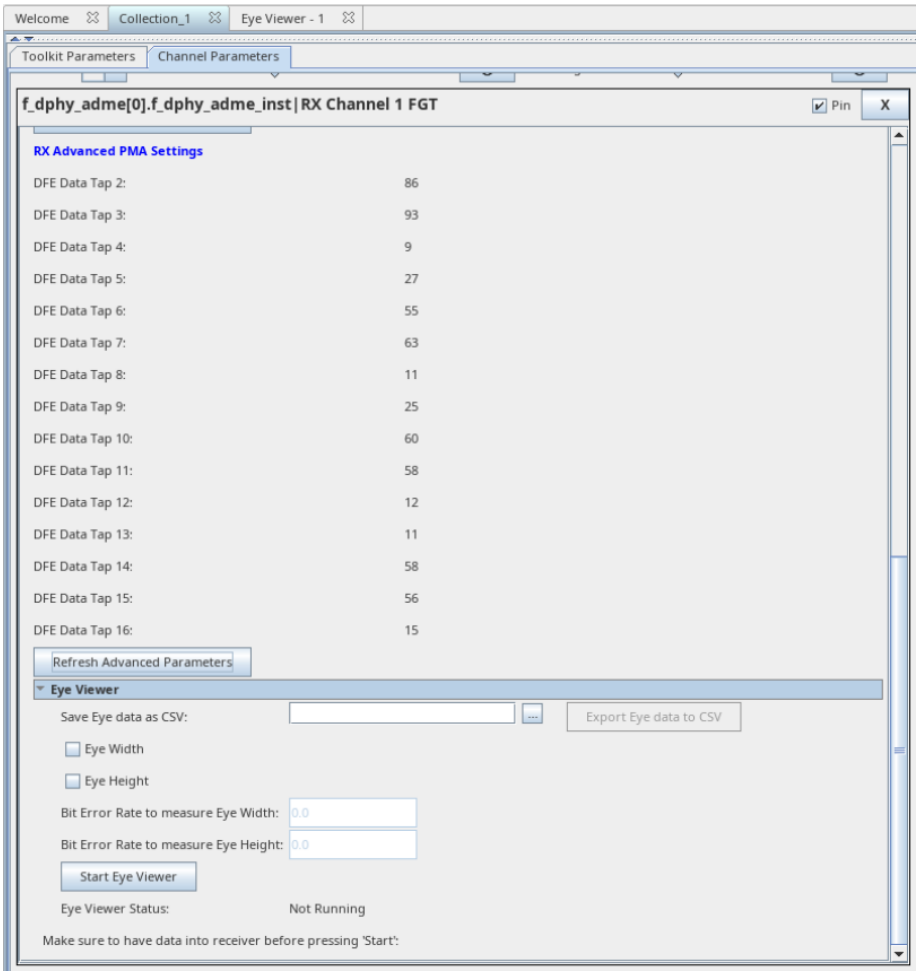
7.2.6. Running Eye Viewer Tests

The Transceiver Toolkit supports internal eye measurements for Intel Agilex 7 F-Tile devices. The Eye Viewer section in every **RX Channel** tab under the **Channel Parameters** pane allows you to set up and run eye measurement tests.

Note: The toolkit does not support 2D Eye plots and it reports the results in terms of **Eye Height** and **Eye Width** values.

1. Select either **Eye Width** or **Eye Height** or both that you want to measure.
2. Set **Bit Error Rate to measure Eye Width** and **Bit Error Rate to measure Eye Height** as shown in the following figure. Valid bit error rate range is from 1.0E-1 to 1.0E-12. The default bit error rate for NRZ is 1.0E-12, and for PAM4 is 1.0E-4.

Figure 127. Setting the BER and Eye Measurement Options



3. Specify the file path to store the results in CSV format.
4. Click **Start Eye Viewer**. Make sure the RX channel is receiving data before starting eye measurement.

When measurement completes, the eye height and eye width results are shown in the following figures. For FGT NRZ signals, the **Eye Center-to-top (Middle)** and **Eye Center-to-bottom (Middle)** values are with reference to the center of the eye. Measurement from eye center to the top of the eye is a positive value. Measurement from the eye center to the bottom of the eye is a negative value. The **Middle Eye Height** is calculated by **Eye Center-to-top (Middle)** minus **Eye Center-to-bottom (Middle)**. Eye width is reported in units of UI and seconds. For PAM4 signals, eye height of three eyes are shown. The **Eye Center-to-top** and **Eye Center-to-bottom** values of the top, middle and bottom eyes are referenced to the center of the middle eye. Measurement above center of the middle eye is a positive value; and

measurement below is a negative value. The **Eye Height** is calculated by **Eye Center-to-top** minus **Eye Center-to-bottom**. A negative **Eye Height** value means the eye is closed. Only the worst eye width of the three eyes is shown.

Figure 128. FGT NRZ Eye Height and Eye Width Results

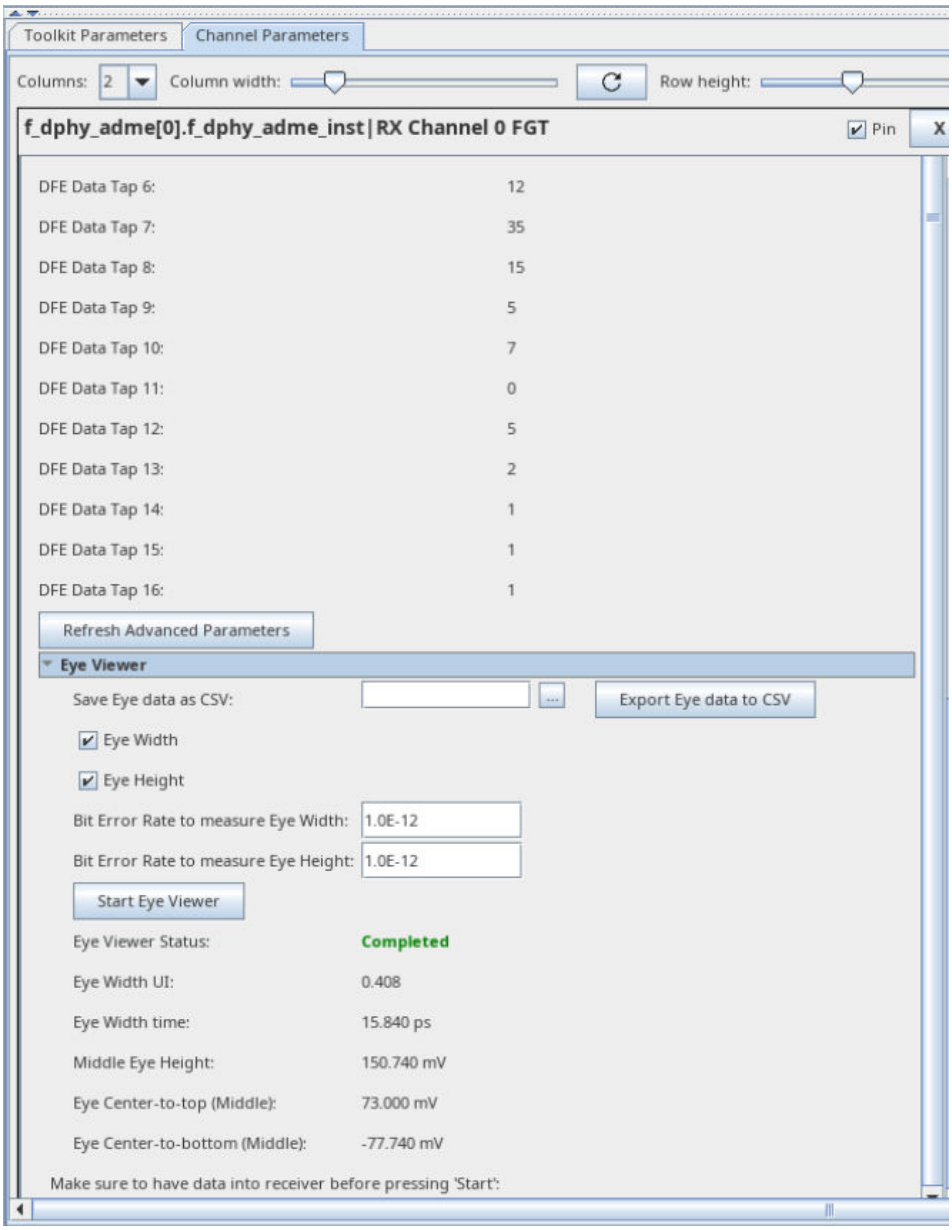
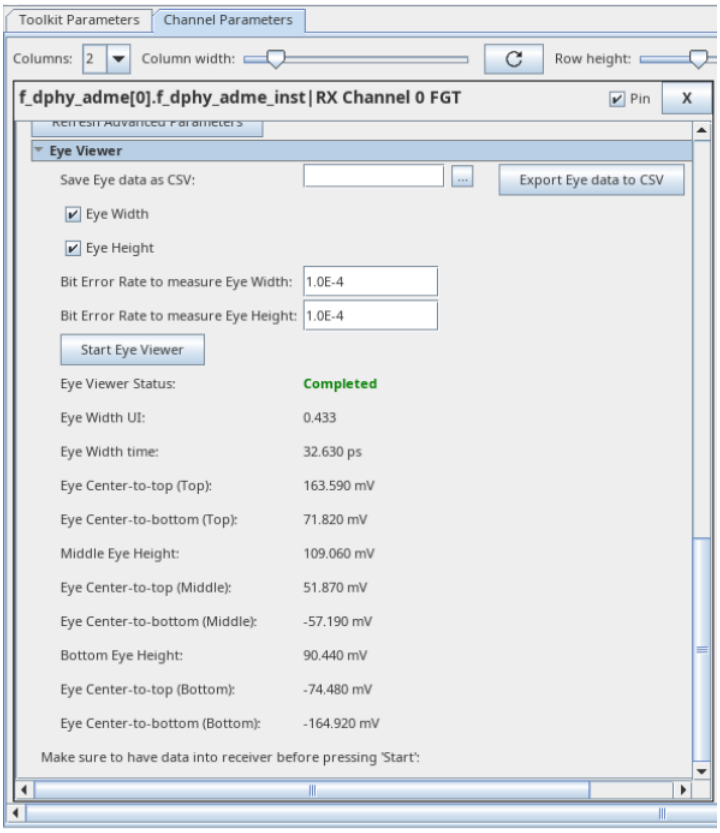


Figure 129. FGT PAM4 Eye Height and Eye Width Results



7.2.7. Running Link Optimization Tests

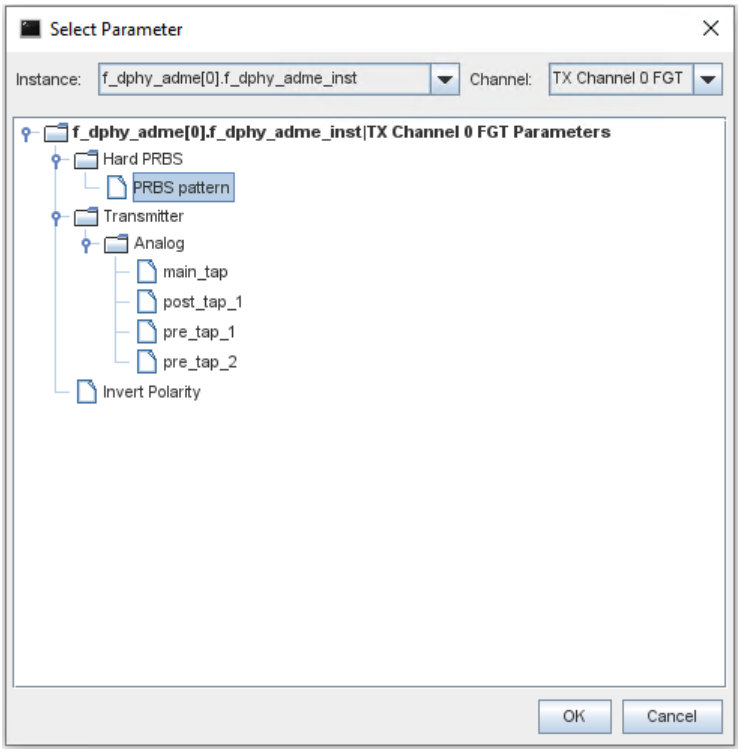
The Transceiver Toolkit autosweep test automatically sweeps PMA setting ranges, PRBS patterns and loopback modes to determine the PMA settings that provide the best signal integrity. The toolkit allows you to store a history of the test runs, and keep a record of the best PMA settings.

To launch autosweep, click **Tools ➤ Autosweep**.

The **Autosweep** view, when launched, is not associated with any given instance(s) or instance or channel pair(s). You can create as many **Autosweep** views as you desire, to allow sweeping over different parameters on different channels of the same instance, or different instances entirely.

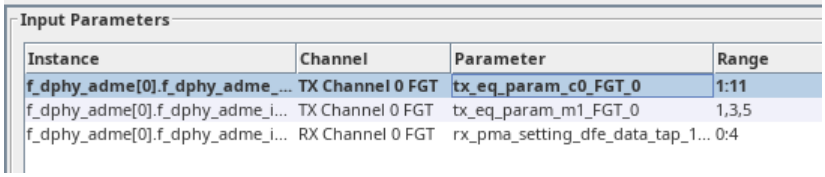
Under the **Input Parameters** pane, click **Add Parameter**. In the **Select Parameter** window, select the instance, channels and parameters to be covered in the autosweep as shown in the following figure.

Figure 130. Select Parameter Window



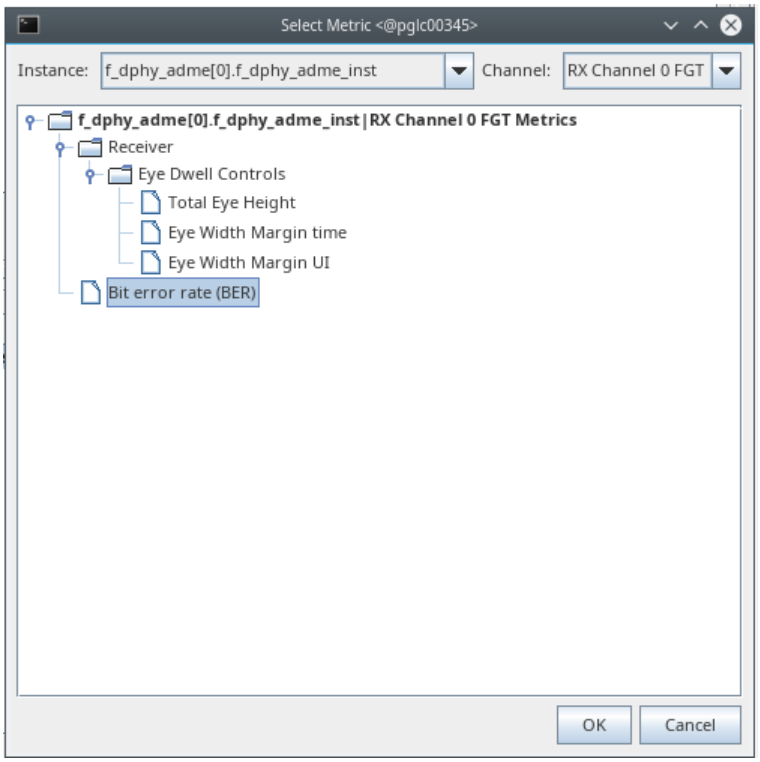
To define the range of values for a parameter, enter the start, and end values separated by a colon symbol. To define specific values, enter the values separated by a comma symbol as shown in the following figure. Place the cursor under the **Range** column of a parameter to show the allowed ranges.

Figure 131. Input Parameters Pane



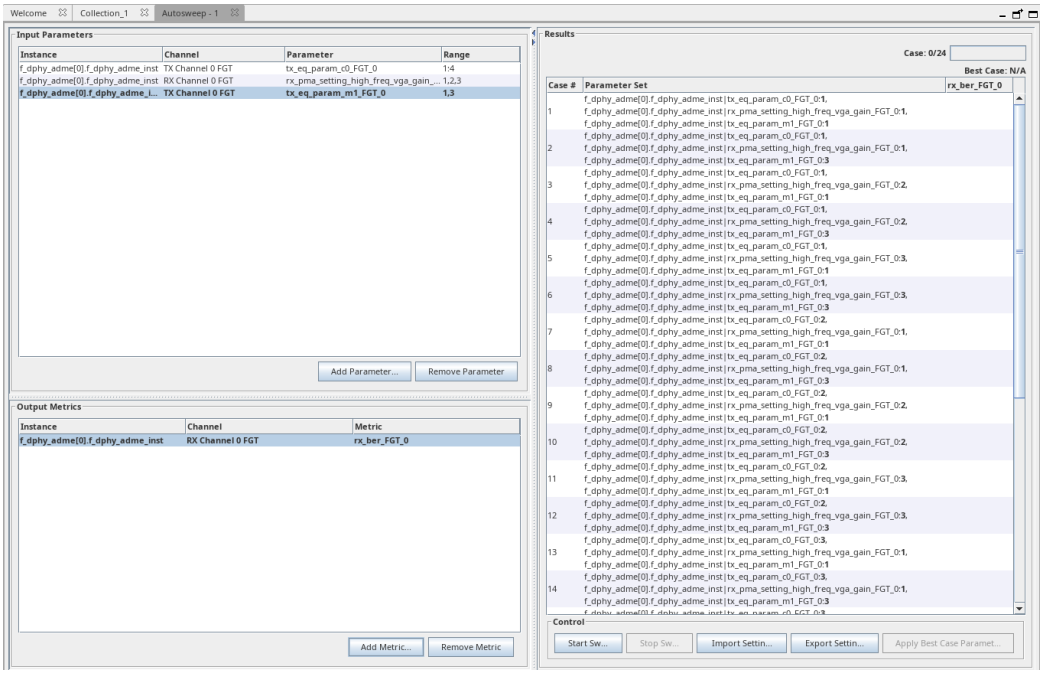
Under **Output Metrics** pane, click **Add Metric**, select **BER**, **Eye Width** or **Eye Height** as the performance metric for each case as shown in the following figure. You can select more than one metric.

Figure 132. Select Metric Window



The following figure shows an example of an **Autosweep** view.

Figure 133. Example of Autosweep View



Under **Control** pane, click **Start Sweep** to start the autosweep. Clicking **Stop Sweep** any time while the autosweep is running, stops the sweep. Click **Export Settings** to save the autosweep setup. Click the **Import Settings** to use a previously saved setup. After the autosweep completes, clicking **Apply Best Case Parameters** populates the parameters of the best case to the selected channel.

7.3. Transceiver Toolkit Parameter Settings

The following table describes the transceiver toolkit parameter settings.

Table 104. Transceiver Toolkit Parameter Settings

Parameter	Description	Control Pane
Auto refresh RX CDR status	Enable this option to update the RX CDR status real time.	Receiver pane.
Auto refresh RX PMA settings	Enable this option to update the RX Equalization settings real time for FGT PMA.	Receiver pane.
Auto refresh TX Status	Enable this option to update the TX PLL lock status real time.	Transmitter pane
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.	Receiver pane
Clear Stats	Clear the current number of bits tested, number of error bits and BER.	Receiver pane
Hard PRBS checker running	Not Running: checker stops. Running: checker is checking, and data pattern is locked.	Receiver pane
Hard PRBS generator running	Not Running: generator stops. Running: generator is sending a pattern.	Transmitter pane
Inject Error	Inject a bit error in the transmitter PRBS pattern.	Transmitter pane
Line encoding	Specifies the modulation type used for serial data.	Transmitter and receiver pane
Loopback mode	Select the loopbacks mode. The available options are: • RX2TXPAR - PMA-receiver-to-transmitter parallel loopback • TX2RXPAR - PMA-transmitter-to-receiver parallel loopback • TX2RXBUF - PMA-transmitter-to-receiver buffer loopback	Transmitter and receiver pane
Number of bits tested	Specifies the number of 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	Receiver pane
Number of error bits	Specifies the number of error bits encountered 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	Receiver pane
PRBS locked	Locked: indicates the PRBS checker is locked to the received PRBS pattern. Not Locked: indicates the PRBS checker is not locked to the received PRBS pattern.	Receiver pane
PRBS pattern	Select the test pattern for the bit error test.	Transmitter and receiver pane

continued...

Parameter	Description	Control Pane
RX CDR locked to ref clock	Locked: Indicates the receiver CDR is in lock-to-reference (LTR) mode. Not Locked: Indicates the receiver CDR is not locked to reference clock. Don't Care: When the receiver CDR is in LTD mode.	Receiver pane
RX CDR locked to data	Locked: Indicates the receiver CDR is in lock-to-data (LTD) mode. Not Locked: Indicates the receiver CDR is not locked to incoming data.	Receiver pane
RX Enable Gray Code	Enables Gray coding for PAM4 only.	Receiver pane
RX PMA Settings	RX Equalization settings.	Receiver pane
RX Polarity Inversion	Enable RX polarity inversion.	Receiver pane
RX Ready	Ready: RX channel out of reset and CDR locks to data. Not Ready: RX channel in reset or CDR is not locked to data.	Receiver pane
RX Reset FGT PMA	Reset the FGT RX datapath. *Note:* Clicking the RX reset of one channel resets all the RX channels in the same F-Tile PMA/FEC Direct PHY Intel FPGA IP instance.	Receiver pane
Start	Starts the pattern generator or checker on the channel to verify incoming data.	Transmitter and receiver pane
Stop	Stops generating patterns and testing the channel.	Transmitter and receiver pane
TX Enable Gray Code	Enables Gray coding for PAM4 only.	Transmitter pane
TX Equalization Parameters	FGT(45) Post_tap_1 Main_tap Pre_tap_1 Pre_tap_2	FHT(46)(47) C_{-3}: Pre-cursor 3 C_{-2}: Pre-cursor 2 C_{-1}: Pre-cursor 1 C_0: Main cursor C_{+1}: Post-cursor 1 C_{+2}: Post-cursor 2 C_{+3}: Post-cursor 3 C_{+4}: Post-cursor 4
TX PLL Locked	Locked: Indicates TX PLL locks to reference clock.	Transmitter pane
TX Polarity Inversion	Enable TX polarity inversion.	Transmitter pane
TX Reset FGT PMA	Reset the FGT TX PMA datapath. *Note:* Clicking the TX reset of one channel resets all the TX channels in the same F-Tile PMA/FEC Direct PHY Intel FPGA IP instance.	Transmitter pane

Related Information

PMA Architecture on page 55

(45) Refer to F-Tile TX Equalizer Tool for legal settings.

(46) Refer to FHT PMA Architecture for legal settings.

(47) When internal serial loopback is enabled, the TX Equalization Parameters are set to default values.

7.4. Troubleshooting Common Errors

Missing High-Speed Link Pin Connections

Check the pin connections to identify high-speed links (tx_p/n and rx_p/n) that are missing. When porting an older design to the latest version of the Intel Quartus Prime software, ensure that these connections exist after porting.

Reset Issues

Ensure that the transceiver channels are not held in reset. You also need to ensure the reset port of the datapath and PMA Avalon memory-mapped interfaces are not held in reset.

Unconnected Clock

The clock inputs of the datapath and PMA Avalon memory-mapped interfaces must be driven with a stable clock within the specific frequency range.

7.5. Transceiver Toolkit Scripts

You can get example scripts with Tcl commands that you can use to access Transceiver Toolkit functions starting from Intel Quartus Prime Pro Edition software version 23.1.

The Transceiver Toolkit scripts allow you to:

- Auto load the design and locate the full path of the toolkit instance in the toolkit.
- Automate the bit-error rate (BER), eye measurement and autosweep tests with predefined settings and collect data while running the script through System Console.
- Control and test multiple channels without the toolkit GUI.
- Display the current progress and result of the test in the System Console **Tcl Console** pane.

Related Information

- Toolkit Tcl Command References
- Analyzing and Debugging Designs with System Console

7.5.1. Supported Transceiver Toolkit Scripts

The transceiver toolkit scripts support FGT and FHT designs. They support both the NRZ and PAM4 modes.

The following table lists the available scripts you can use:

Table 105. Transceiver Toolkit Scripts

Tcl File Name	Description
`device_initialization.tcl`	You must update this script with which device is on the JTAG chain and what `.sof` is programmed into the device. When you run this script, it auto-loads the toolkit and automatically link the design information to the connected device.

`fgt_nrz_ber_eye_test.tcl`
`fgt_pam4_ber_eye_test.tcl`
`fht_nrz_ber_eye_test.tcl`
`fht_pam4_ber_eye_test.tcl`
	These scripts are used to run BER and Eye measurement tests. Select the scripts based on your design mode. The available design modes are FGT NRZ, FGT PAM4, FHT NRZ and FHT PAM4 designs.

`fgt_nrz_autosweep_test.tcl`
`fgt_pam4_autosweep_test.tcl`
`fht_nrz_autosweep_test.tcl`
`fht_pam4_autosweep_test.tcl`
	These scripts are used to run the Autosweep test. Select the scripts based on your design mode. The available design modes are FGT NRZ, FGT PAM4, FHT NRZ and FHT PAM4 designs.

`unload_toolkit.tcl`	Run this script to close the toolkit after you have finished all the tests.

When you install Intel Quartus Prime Pro Edition software version 23.2 or later, the template of the scripts is included in the installation directories. The following table lists the location of the Transceiver Toolkits scripts:

Table 106. Script Location

Platform	Path
Windows	`<drive>:\intelFPGA_pro\23.2\ip\altera\alt_xcvr\toolkit\ftile_transceiver_toolkit\scripts\`
Linux	`<path to acds folder>/ip/altera/alt_xcvr/toolkit/ftile_transceiver_toolkit/scripts`

7.5.2. Modifying the Scripts

There are several variables inside the script, such as setting up the channel links, PRBS pattern, loopback mode, BER test duration, and the TX equalizer settings that you must modify based on your design configuration. You must set the values of these variables in the script before running the test. The following procedures describes the steps to modify the script for each test:

A. Modify the Script for Device Initialization and Toolkit Bring-Up

You must modify the device initialization script and update the device that are on the JTAG chain and point to the `.sof` that you have programmed.

1. Open `device_initialization.tcl` script in any text editor.
2. Make the following modifications to the `device_initialization.tcl` script:

```
set path <sof-file-path>
set result_dir <my_diretory>
set device_die_name <device-name>
```

An example of the settings in the `device_initialization.tcl` script is shown in the following figure.

Figure 134. Device Initialization Settings

```
set path top.sof
set result_dir C:/Users/Apps_xrio/results
set device_die_name "AGIB027R29AR"
```

Note: If you are using the Windows platform, use the `.sof` file name instead of sof file path in step 2 above.

B. Modifying the Script to Run BER and Eye Measurement Tests

Refer to the *Transceiver Toolkit Scripts* table to select the Transceiver Toolkit script for your desired mode.

1. Choose the script based on your design mode and open it in any text editor.
2. Make the following modifications to the script:
   - Set 1 to enable the test you want to run as shown below:

```
############### Tests to run: 0 = bypass, 1 = run ###############
set run_ber_test 1
set run_eye_test 0
```

   - Set up the test variables. There are six main variables that you need to modify in the `link_test_parameters` list in the script:
     — TX logical channel (index 0)
     — RX logical channel (index 1)
     — PRBS pattern (index 2 and 3)
     — Loopback mode (index 4)
     — TX and RX PMA settings (index 5 to 11)
     — Eye measurement settings (index 12 to 15)

     Note: Currently the Transceiver Toolkit only supports RX PMA auto-adaptation mode. You can leave the RX PMA settings to the default values.

     For example, if you want to run the BER test between TX channel 0 and RX channel 0, set both index 0 and index 1 to value 0. You can also link TX and RX channels in different physical channel locations. For example, to link TX channel 0 to RX channel 1, set index 0 to value 0 and index 1 to value 1. In order to link TX and RX channels with different physical locations, make sure you have an external loopback, either through a loopback cable or card on the board. An example of the `link_test_parameters` settings is shown below:

```
#####################################################
###      Customize the test variable              ###
#####################################################
# The list_test_parameters' indexing :
#   index 0 - TX Logical Channel
#   index 1 - RX Logical Channel
#   index 2 - TX PRBS Generator Pattern :
#             PRBS7,PRBS9,PRBS10,PRBS13,PRBS15,PRBS23,PRBS28,
#              PRBS31,QPRBS13,PRBS13Q,PRBS31Q,SSPR,SSPR1,SSPRQ
#   index 3 - RX PRBS Checker Pattern : PRBS7,PRBS9,PRBS10,PRBS13,
#             PRBS15,PRBS23,PRBS28,PRBS31,QPRBS13,PRBS13Q,PRBS31Q,
#             SSPR,SSPR1,SSPRQ
#   index 4 - Loopback Mode : PMA TX to RX Buffer lbpk - "TX2RXBUF"
#                           ; PMA TX to RX parallel lpbk  - "TX2RXPAR"
#                           ; PMA RX to TX parallel lpbk  - "RX2TXPAR"
#   index 5 - TX Pre-Tap 2  :  {0 to 7}
```

```
#   index 6 - TX Pre-Tap 1  :  {0 to 15}
#   index 7 - TX Main Tap   :  {0 to 46}
#   index 8 - TX Post-Tap 1 :  {0 to 19}
#   index 9 - RX High Freq VGA Gain  :  {0 to 127}
#   index 10 - RX High Freq Boost  :  {0 to 63}
#   index 11 - RX DFE Data Tap 1   :  {0 to 63}
#   index 12 - Enabling the eye height test : Enable - "true"  ;
#                                             Disable - "false"
#   index 13 - Set the Bit Error Rate to measure Eye Height :
#              Min - 1.0E-1  Maximum - 1.0E-12
#   index 14 - Enabling the eye width test :
#              Enable - "true"  ;  Disable - "false"
#   index 15 - Set the Bit Error Rate to measure Eye Width :
#              Min - 1.0E-1 Maximum - 1.0E-12

set link_test_parameters {{0 0 "PRBS23" "PRBS23" "TX2RXBUF" "0" "0" "0"
  "0" "0" "0" "0" "true"  "1.0E-4" "true" "1.0E-4"}
                          {1 1 "PRBS9"  "PRBS9"  "TX2RXBUF" "0" "0" "0"
  "0" "0" "0" "0" "true"  "1.0E-4" "true" "1.0E-4"}}
```

Note: The example above depicts the settings for two links only. If you want to run the test for four links, you have to add two more rows in the `link_test_parameters` list accordingly.

- You must set the criteria to stop the BER test for a channel, either based on maximum error bits or the BER test duration. The following code shows an example on how to have the BER test stop after 6 seconds and displays the status of the link every 2 seconds. The `max_error_bits` indicates the maximum number of error bits encountered by the link. The test stops after reaching the maximum number of error bits even if the maximum run time of 6 seconds has not elapsed.

```
############### Setup link run length ############################
set max_error_bits 10
set max_run_time_in_seconds 6
set checker_status_polling_interval_in_seconds 2
```

C. Modifying the Script to Run Autosweep Test

1. Choose the script based on your design mode and open it in any text editor.
2. Make the following modifications to the script:
   - Set up the test variables for TX and RX PMA sweep range. There are seven main variables you need to modify in the `link_test_parameters` list in this script:
     — TX logical channel (index 0)
     — RX logical channel (index 1)
     — PRBS pattern (index 2 and 3)
     — Loopback mode (index 4)
     — Input parameters (index 5 to 11)
     — Output metric (index 12 to 17)
     — BER test duration (index 18)

     For example, to add BER as the output metric, set index 12 to value 1 as shown below:

```
#########################################################
###      Customize the test variable              ###
#########################################################
```

```
# The list_test_parameters' indexing :
#   index 0 - TX Logical Channel
#   index 1 - RX Logical Channel
#   index 2 - TX PRBS Generator Pattern : PRBS7,PRBS9,PRBS10,PRBS13,
#             PRBS15,PRBS23,PRBS28,PRBS31,QPRBS13,PRBS13Q,PRBS31Q,
#             SSPR,SSPR1,SSPRQ
#   index 3 - RX PRBS Checker Pattern : PRBS7,PRBS9,PRBS10,PRBS13,
#             PRBS15,PRBS23,PRBS28,PRBS31,QPRBS13,PRBS13Q,PRBS31Q,
#             SSPR,SSPR1,SSPRQ
#   index 4 - Loopback Mode : PMA TX to RX Buffer lbpk - "TX2RXBUF"
#                           ; PMA TX to RX parallel lpbk  - "TX2RXPAR"
#                           ; PMA RX to TX parallel lpbk  - "RX2TXPAR"
#   index 5 - TX Pre-Tap 2  :  {0 to 7}  if you put like 0:5 it
#             autosweeps the channel for each number start from 0 to 5
#   index 6 - TX Pre-Tap 1  :  {0 to 15}
#   index 7 - TX Main Tap   :  {0 to 46}    0,1.5 0,5,6
#   index 8 - TX Post-Tap 1 :  {0 to 19}
#   index 9 - RX High Freq VGA Gain  :  {0 to 127}
#   index 10 - RX High Freq Boost  :  {0 to 63}
#   index 11 - RX DFE Data Tap 1   :  {0 to 63}
#   index 12 - Adding BER matric :  "1"  ;  Disable - "0"
#   index 13 - Adding total height measurement matric :
#              Enable - "1"  ;  Disable - "0"
#   index 14 - Adding eye width time matric :
#              Enable - "1"  ;  Disable - "0"
#   index 15 - Adding eye width UI matric :
#              Enable - "1"  ;  Disable - "0"
#   index 16 - Extrapolarate Rate:
#              Min - 1E-1  ;  Max - 1E-12
#   index 17 - Extrapolarate Width Rate:
#              Min - 1E-1  ;  Max - 1E-12
#   index 18 - BER test duration per case (seconds): Max 1E3

set link_test_parameters {{0 0 "PRBS23" "PRBS23" "TX2RXBUF" "0:0"
"0:1" "0:1" "0:0" "0:0" "0:0" "0:0" "1" "0" "0" "0" "1e-4" "1e-4" "2" }
                          {1 1 "PRBS10" "PRBS10" "TX2RXBUF" "0:0"
"0:1" "0:0" "0:0" "0:0" "0:0" "0:0" "1" "0" "0" "0" "1e-4" "1e-4" "2" }}
```

- You can get the Autosweep test data in xml format. The command shown below is in the Autosweep script and you can change the file name and result directory to your own file name and directory.

```
autosweep_get_data -outputfile <result-directory>/<file-name>.xml
$autosweep_inst_id
```

7.5.3. Script Execution

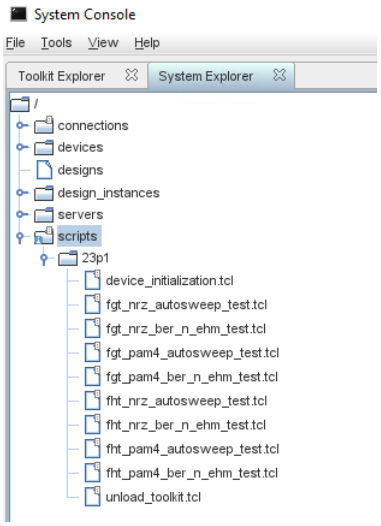
You can execute the scripts either by saving the script in your project directory or in the **scripts** folder under the **System Explorer** pane. The following steps describe on how you can execute the scripts to run the Transceiver Toolkit test:

1. Click **Tools > Programmer**. Configure your device with the `.sof` file.
2. Save the scripts that you modified in *Modifying the Scripts* section into your project directory or into the scripts folder under **System Explorer** pane. The path to save the scripts under the **System Explorer** pane in **scripts** folder is shown below.

   ```
   Windows: <drive>:\Users\<username>\system_console\scripts\
   Linux: <$HOME > system_console/scripts
   ```

3. If you save the Tcl files into the **scripts** folder under **System Explorer** pane; to execute the script, click **Tools > System Debugging Tools > System Console**. In the **System Explorer** pane double click the scripts folder, then double click the script that you want to execute.

Figure 135. System Explorer Pane



4. If you save the Tcl files into your project directory, change directory to your project directory and source the script in the Tcl console pane. For example:

```
cd <my_project_dir>
source device_initialization.tcl
```

Note: You must modify the scripts first before you execute them to run the tests. The channel PMA settings, PRBS pattern, loopback mode must be set up in the scripts. Refer to Modifying the Scripts for more information.

Related Information

System Explorer Pane

7.5.4. Example of the Results in Tcl Console

While running the test, you can observe the progress of the test and display the results in **Tcl Console**. The following is an example of a FGT PAM4 design test results that you can observe in **Tcl Console**.

1. Device initialization output.

Figure 136. Device Initialization Result

```
------------------- Initialization ----------------------

Device: Path = /devices/AGIB027R29AR(0|1|2|3)@1#USB-1#Agilex-I F-Tile Development Kit#pg-ceglab02.gar.corp.intel.com

%
```

2. BER test results for two channels.

Figure 137. BER Test Results

```
Data rate (Mbps): 53125
Line Encoding: PAM4



Running BER test for 6 seconds with the following settings:
 TX Logical Channel: 0
 RX Logical Channel: 0
 TX PRBS   : PRBS23
 RX PRBS   : PRBS23
 Lock to data: Locked
 Checker is running

Elapsed time(sec)   Total bits    Error bits  Bit error rate
================= ============= ============ ==============
                2   241267939656            0             0.0
                4   347321034222            0             0.0
                6   453546588867            0             0.0

Test stopped after achieving maximum run time of 6 seconds


Running BER test for 6 seconds with the following settings:
 TX Logical Channel: 1
 RX Logical Channel: 1
 TX PRBS   : PRBS9
 RX PRBS   : PRBS9
 Lock to data: Locked
 Checker is running

Elapsed time(sec)   Total bits    Error bits  Bit error rate
================= ============= ============ ==============
                2   241139215230            0             0.0
                4   347235488146            0             0.0
                6   455289188027            0             0.0

Test stopped after achieving maximum run time of 6 seconds


Finished
Run the 'unload_toolkit.tcl' if you have finished all the tests.
```

3. Eye width and eye height test results for two channels.

Figure 138. Eye Width and Eye Height Test Results

```
Data rate (Mbps): 53125
Line Encoding: PAM4


Running eye measurement with the following settings:
    TX Logical Channel: 0
    RX Logical Channel: 0
    TX PRBS           : PRBS23
    RX PRBS           : PRBS23
    Lock to data      : Locked
    Checker           : Running

Running Eye Test.....

Eye Width Margin UI Eye Width Margin Time
=================== ====================
          0.517         19.453 ps

       Eye-Height  Eye-Centre-to-Top  Eye-Centre-to-Bottom
====== ==========  =================  ====================
Top     74.480 mV      152.950 mV         78.470 mV
Middle  82.460 mV       39.900 mV        -42.560 mV
Bottom  75.810 mV      -85.120 mV       -160.930 mV


Running eye measurement with the following settings:
    TX Logical Channel: 1
    RX Logical Channel: 1
    TX PRBS           : PRBS9
    RX PRBS           : PRBS9
    Lock to data      : Locked
    Checker           : Running

Running Eye Test.....

Eye Width Margin UI Eye Width Margin Time
=================== ====================
          0.367         13.805 ps

       Eye-Height  Eye-Centre-to-Top  Eye-Centre-to-Bottom
====== ==========  =================  ====================
Top     65.170 mV      146.300 mV         81.130 mV
Middle  81.130 mV       47.880 mV        -33.250 mV
Bottom  67.830 mV      -70.490 mV       -138.320 mV


Finished
Run the 'unload_toolkit.tcl' if you have finished all the tests.

% |
```

4. Autosweep results for one channel.

Figure 139. Autosweep Test Results

```
Data rate (Mbps): 53125
Line Encoding: PAM4


Running Autosweep with the following settings:
  TX Logical Channel        : 0
  RX Logical Channel        : 0
  TX PRBS                   : PRBS23
  RX PRBS                   : PRBS23
  Lock to data              : Locked
  Checker is running
  Pre-emphasis main tap     : 0:1
  Pre-emphasis 1st pre-tap  : 0:1
  Pre-emphasis 2nd pre-tap  : 0:0
  Pre-emphasis 1st post-tap : 0:0
  High Frequency VGA Gain   : 0:0
  High Frequency Boost      : 0:0
  DFE Data Tap 1            : 0:0

 Total number of case : 4
 Autosweep test is in progress...

 Current progress... 0 %
 Current progress... 25 %
 Current progress... 50 %
 Current progress... 75 %
 Current progress... 100 %
 Complete
 BER test duration per case : 2

 Case #        BER             Total Eye Height              Eye witdh UI      Eye width Time (ps)
======= ============= ============================== ================== ====================
   1           0.0       75.810 mV 86.450 mV 77.140 mV           0.533               20.08
   2           0.0       78.470 mV 85.120 mV 77.140 mV           0.500               18.82
   3           0.0       78.470 mV 89.110 mV 81.130 mV           0.633               23.84
   4           0.0       74.480 mV 85.120 mV 74.480 mV           0.816               30.74
```

8. F-tile Architecture and PMA and FEC Direct PHY IP User Guide Archives

For the latest and previous versions of this user guide, refer to the *F-tile Architecture and PMA and FEC Direct PHY IP User Guide*. If an IP or software version is not listed, the user guide for the previous IP or software version applies.

9. Document Revision History for F-tile Architecture and PMA and FEC Direct PHY IP User Guide

Document Version	Intel Quartus Prime Version	Changes
2023.06.26	23.2	Made the following changes: • Removed redundant figure from *FGT-PMA-to-400G-Hard-IP-Fracture Mapping* section. • Removed the list of OPNs and updated *FGT Transmitter Buffer and Phase Generator* section with OPN list link in *Appendix A.1*. • Added new section *FGT Reference Clock Receiver Analog Front End* describing termination requirements for the FGT reference clock. • Updated description for the **Enable fgt_rx_cdr_fast_freeze_sel port** and **Enable fgt_rx_cdr_freeze port** parameters in the *RX FGT PMA Parameters* table. • Added note about how to access the CWBIN and FEC registers in the *RS-FEC (Reed Solomon Forward Error Correction) Options* section. • Added note in description column of *TX and RX Reference Clock and Clock Output Interface Signals* table for `rx_clkout, rx_clkout2, tx_clkout, tx_clkout2` signals. • Updated description for `fgt_rx_cdr_freeze[N-1: 0]` signal and added new signal `fgt_rx_cdr_fast_freeze_sel[N-1: 0]` in *RX PMA Status Signals* table. • Added frequency range for `reconfig_xcvr_clk` and `reconfig_xcvr<n>_clk` clock signals in *PMA Avalon Memory Mapped Interface Signals* section. • Added frequency range for `reconfig_pdp_clk` and `reconfig_pdp<n>_clk` clock signals in *Datapath Avalon Memory Mapped Interface Signals* section. • Removed the *FGT Core PLL Mode* section as the mode is no longer supported in the Intel Quartus Prime Pro Edition software. • Reorganized the content in the *Configurable Intel Quartus Prime Software Settings* section. • Updated the *FGT PMA Settings* section with valid parameter settings for *TX Equalization* and *RX Manual Equalization* parameters. • Updated the *Direct Register Method Examples* section with *Measuring the Bit Error Rate (BER) with FGT PMAs* example and a note for *Internal Serial Loopback* and *Polarity Inversion* examples. • Updated the *FGT Attribute Access Data Value 2* table with *Get Status* opcodes. • Removed parameter **Refclk is available at device configuration** from the *F-Tile Reference and System PLL Clocks Intel FPGA IP Parameters* table. • Updated the *F-Tile Reference and System PLL Clocks Intel FPGA IP Parameters* table with new parameters **Refclk #n is active at and after device configuration** and **Export Refclk #n for use in user logic** • Updated the *F-Tile Reference and System PLL Clocks Intel FPGA IP Port List* table with the following signals; `avmm_clk`, `avmm_reset`, `refclock_ready [2:0]`, `en_refclk_fgt_i[1:0]`, `refclk_fgt_enabled_i`, and `out_coreclk_i`.

continued...

Document Version	Intel Quartus Prime Version	Changes
		• Replaced section *Guidelines to Indicate all System PLL Reference Clocks are Ready* with new section *Guidelines for Refclk #i is Active At and After Device Configuration*. • Replaced section *Example of Reference Clock Availability at Device Programming* with new section *Guidelines for System PLL Reference Clock*. • Replaced section *Example Flow to Indicate All System PLL Reference Clocks are Ready* with new section *Guidelines for FGT Reference Clock*. • Removed the chapter *Implementing the F-Tile Global Avalon Memory-Mapped Interface Intel FPGA IP* from the user guide as the IP is no longer supported in the Intel Quartus Prime Pro Edition software. • Removed the list of OPNs and updated *F-Tile TX Equalizer Tool* section with OPN list link in *Appendix A.1*. • Updated the *Tcl File Name* and *Path* of scripts in the *Supported Transceiver Toolkit Scripts* section. • Added *Appendix A.1* with list of *F-tile Production Revision and Firmware OPNs*.
2023.04.03	23.1	Made the following changes: • Updated the product family name to *Intel Agilex 7*. • Added new information about mixed transceiver mode (**topology 6a**) support in *400G Hard IP and 200G Hard IP*, *PMA-to-Fracture Mapping*, and *Topologies* sections. • Updated *Fracture Type Used by Mode* table with 200GbE-4 support in **st_x16** row. • Updated *Clock Rules and Restrictions* section with additional requirements for a stable reference clock. • Updated *FGT Transmitter PMA Equalizer Parameters for NRZ and PAM4 Modes* tables with additional information and missing default values. • Updated *FGT Data Pattern Generator and Verifier* section with PRBS and SSPR specification information. • Updated *FGT PMA Loopback Modes* section with additional information about the various loopback modes. • Updated *FGT PMA Loopback Modes* figure with missing connection. • Added description about the 32-bit soft CWBIN counters in *RS-FEC (Reed Solomon Forward Error Correction) Options* section. • Added new parameters **Include 32bit soft CWBIN counters** and **Reconfig clock frequency** in *RS-FEC Parameters* table for soft CWBIN counters support. • Updated *FGT PMA Fractional Mode* section with corrected information about setting the fractional mode. • Added new section *Accessing Configuration Registers* with detailed information about how to use the offset address in the F-Tile PMA/FEC Direct PHY Intel FPGA IP register map to access the registers. • Corrected *Configurable Intel Quartus Prime Software Settings* section with FGT TX equalization example `qsf` values for `txeq_pre_tap_1` and `txeq_post_tap_1` and added a note in the section. • Updated the bypass RX auto adaptation `qsf` assignments and added the RX manual equalization `qsf` assignments in *Configurable Intel Quartus Prime Software Settings* section. • Corrected *Direct Register Method Example* section with TX equalizer pre-cursor 1 register and TX equalizer post-cursor 1 register addresses for FGT PMA. • Added additional FGT PMA setting examples in *Direct Register Method Examples* section. • Added note in *FGT Attribute Access Method Example 2* section about register 0x90040[25:24] status values. • Updated *F-Tile Reference and System PLL Clocks Intel FPGA IP Parameters* table with renamed **Refclk is available at device configuration** parameter.

continued...

Document Version	Intel Quartus Prime Version	Changes
		• Updated *Example Flow to Indicate All System PLL Reference Clocks are Ready* section with **Refclk is available at device configuration** parameter renaming. • Added **Enable Debug Master Endpoint on Global AVMM** parameter to the *F-Tile Global Avalon Memory-Mapped Interface Intel FPGA IP Parameters* table. • Updated *Hardware Flow Using the F-Tile Global Avalon Memory-Mapped Interface Intel FPGA IP* section with usage information about **Enable Debug Master Endpoint on Global AVMM** parameter. • Added note in *Running Eye Viewer Tests* section about no support for 2D Eye plots. • Added new section *Transceiver Toolkit Scripts* with information about using scripts available in the Intel Quartus Prime Pro Edition software version 23.1 for transceiver testing.
2023.01.25	22.4	Updated the *FHT Loopback Mode* figure with the correct Deserializer block location.
2022.12.19	22.4	Made the following changes: • Updated *Clock Rules and Restrictions* section with additional information about the reference clocks and a new note. • Updated *FGT Transmitter PMA Equalizer Parameters for NRZ and PAM4 Modes* table; renamed cursor to tap, and updated the footnotes. • Added equations based on device OPN list for the transmitter buffer equalizer parameters in the *FGT Transmitter Buffer and Phase Generator* section. • Corrected the preset naming in the *F-Tile PMA/FEC Direct PHY Intel FPGA IP Available Parameter Presets* table in *Preset IP Parameter Settings* section. • Updated the description for the **PMA parallel clock frequency** parameter in *General and Common Datapath Options* table. • Updated the *Show Preset Settings* figure in *Example Design Generation* section. • Corrected `tx_coreclkin` and `rx_coreclkin` signal names in *Recommended tx/rx_coreclkin Connection and tx/rx_clkout2 Source* section. • Corrected the *k* counter descriptions and equations in *FGT PMA Fractional Mode* section. • Added new information about meeting jitter specifications for OTN/SDI and in other modes in *FGT PMA Fractional Mode* section. • Updated `rx_ready` signal description in *Reset Signal Descriptions* table. • Updated the steps and figure in the *Run-time Reset Sequence—TX* section. • Updated the steps and figure in the *Run-time Reset Sequence—RX* section. • Updated the steps and figure in the *Run-time Reset Sequence—TX + RX* section. • Updated column header for *FHT PMA Lane Number and Offset Address* and *FGT PMA Lane Number and Offset Address* tables in *Lane Offset Address* section. • Updated descriptions for calculating the incremental lane numbers for the FHT PMA and FGT PMA in *Lane Offset Address* section. • Added `.qsf` settings to bypass RX auto adaptation in *Configurable Intel Quartus Prime Software Settings* section. • Updated *Guidelines for F-Tile Reference and System PLL Clocks Intel FPGA IP Usage* with additional information about the system PLL reference clock.

continued...

Document Version	Intel Quartus Prime Version	Changes
		• Updated *Guidelines to Indicate all System PLL Reference Clocks are Ready* with additional information about using the internal clock to calibrate and configure the device, information about PCIe specification compliance, and OPN device list that supports **Refclk is available at power-on** parameter. • Added additional information in *Example of Reference Clock Availability at Device Programming* section to clarify the example. • Added tip for step 5b. in *Hardware Flow Using the F-Tile Global Avalon Memory-Mapped Interface Intel FPGA IP*. • Added new section *Examples of Register Access Using the F-Tile Global Avalon Memory-Mapped Interface Intel FPGA IP* with several topics with examples. • Updated the FGT transmitter equalizer cursor naming, added additional tool link for various devices based on OPNs, and updated figure in *F-Tile TX Equalizer Tool* section. • Updated the FGT transmitter equalizer cursor naming, updated descriptions for **RX Ready** and **PRBS locked** parameters in *Transceiver Toolkit Parameter Settings* table.
2022.11.03	22.3	Added a clarifying note about the SATA and USB protocol modes in sections: *FGT PMA Configuration Rules for SATA and USB mode* and *TX Parallel Data Mapping Information for SATA and USB Protocol Modes for Different Configurations*.
2022.09.26	22.3	Made the following changes: • Updated *FGT Transmitter PMA Equalizer Parameters for NRZ and PAM4 Modes* table with total slice values for different OPNs. • Updated `tx_reset_ack` and `rx_reset_ack` signal descriptions in the *Reset Signals* and *Reset Signal Descriptions* tables. • Added new parameters **FGT PMA configuration rules** and **Enable simplified TX data interface** in *General and Common Datapath Options* table. • Added new section *FGT PMA Configuration Rules for SATA and USB mode.* • Added new parameters **Enable fgt_tx_beacon port** and **Enable Spread Spectrum clocking** in *TX FGT Datapath Parameters* table for SATA and USB support. • Added new parameters **Enable fgt_rx_cdr_fast_freeze_sel port** and **Enable fgt_rx_cdr_set_locktoref port** in *RX FGT PMA Parameters* table for GPON support. • Updated descriptions for **Enable fgt_rx_signal_detect port** and **Enable fgt_rx_signal_detect_lfps port** parameters in *RX FGT PMA Parameters* table for SATA and USB support. • Corrected *TX PMA Status Signals* to *TX PMA Control Signals* and added new signal `fgt_tx_pma_elecidle` to the table. • Added new section *TX Parallel Data Mapping Information for SATA and USB Protocol Modes for Different Configurations*. • Updated and replaced PMA Avalon memory-mapped with Global Avalon memory-mapped interface in *Guidelines to Indicate all System PLL Reference Clocks are Ready* section and the examples. • Added additional examples in step 6. of *Hardware Flow Using the F-Tile Global Avalon Memory-Mapped Interface Intel FPGA IP* section. • Updated *Simulating the F-Tile PMA/FEC Direct PHY Design* section with information about the auto-generated file names.
2022.06.24	22.2	Made the following changes: • Updated *F-Tile Supported FEC Modes and Compliance Specifications* table with Fibre Channel 64G support and added note. • Clarified fourth bullet in *Clock Rules and Restrictions* section with updated rules for the FHT microcontroller reference clock. • Added new figure for TX termination in *FGT Transmitter Buffer and Phase Generator*.

continued...

Document Version	Intel Quartus Prime Version	Changes
		• Added new figure for RX termination in *FGT Receiver Buffer and Equalizer*. • Added new topic *Register Map IP-XACT Support* in *Configuring the IP* section. • Added instructions to simulate example design using VCS MX and Xcelium simulators in *Example Design Simulation* section. • Added note for `tx_pll_refclk_link` and `rx_cdr_refclk_link` signals in the *TX and RX Reference Clock and Clock Output Interface Signals* table. • Updated *FGT Attribute Access Method* section with additional information and tables. • Added a new example for serial loopback enable and disable in *FGT Attribute Access Method* section. • Deleted note in step 7. of *FGT Attribute Access Method Example 2*. • Added new parameter **Refclk is available at power-on** in *Implementing the F-Tile Reference and System PLL Clocks Intel FPGA IP* chapter. • Added new section *Guidelines to Indicate all System PLL Reference Clocks are Ready* in *Implementing the F-Tile Reference and System PLL Clocks Intel FPGA IP* chapter. • Updated step 5. of *Hardware Flow Using the F-Tile Global Avalon Memory-Mapped Interface Intel FPGA IP*. • Added FEC alignment marker information and table in *Implementing a RS-FEC Direct Design in the F-Tile PMA/FEC Direct PHY Intel FPGA IP* • Added data scrambling and de-scrambling information for FEC in *Implementing a RS-FEC Direct Design in the F-Tile PMA/FEC Direct PHY Intel FPGA IP*. • Updated *Running Eye Viewer Tests* with an additional step and figures. • Added information about **Autosweep** in *Running Link Optimization Tests*.
2022.03.28	22.1	Made the following changes: • Clarified third bullet in *Clock Rules and Restrictions* section with updated rules for the system PLL reference clock. • Added note about bonding rules in the *Bonding Placement Rules* section. • Added new topic *Tuning the Fractional Value in Fractional Mode* in the *FGT PMA Fractional Mode* section. • Added *PMA Reconfiguration Interface* column in the *Reset Signals—Block Level* table and added a note about the `reconfig_xcvr_reset` signal usage. • Updated TX equalization main tap example setting for *FHT PMA* in *Configurable Intel Quartus Prime Software Settings* section. • Added *FGT Attribute Access Method Example 2* in *FGT Attribute Access Method* section for TX and RX polarity inversion. • Updated *Running BER Tests* section with information about the **Actions** sub-menu. • Updated *Running Eye Viewer Tests* section with detailed description about using the **Eye Viewer** tool for eye height measurement for the FGT PMA.
2021.12.15	21.4	Made the following changes: • Updated *Increment and Decrement Size* column in the *FGT Transmitter PMA Equalizer Parameters for NRZ and PAM4 Modes* table. • Added footnote for *System PLL frequency* description in *General and Common Datapath Options* table. • Added footnote for *TX FGT PLL reference clock frequency* description in *TX FGT Datapath Parameters* table. • Added footnote to *TX and RX Reference Clock and Clock Output Interface Signals* table.

continued...

Document Version	Intel Quartus Prime Version	Changes
		• Removed note for *Enable rx_cdr_divclk_link0 port* and *Enable rx_cdr_divclk_link1 port* parameters in the *RX FGT PMA Parameters* table. • Updated *Example Design Generation* with RS-FEC example design information. • Added note to *Clocking* section in *Implementing the F-Tile PMA/FEC Direct PHY Intel FPGA IP* chapter. • Added *RX invert P and N*, *RX termination*, *TX invert P and N*, *TX termination*, *TX out tristate enable* and *TX equalization* `qsf` settings for *FHT PMA* in *Configurable Intel Quartus Prime Software Settings* section. • Removed *RX termination mode select* `qsf` setting for *FGT PMA* in *Configurable Intel Quartus Prime Software Settings* section. • Added *TX equalization* `qsf` setting for *FGT PMA* in *Configurable Intel Quartus Prime Software Settings* section. • Updated steps 4a, 6, 8a, 10c, 13c, 14b and 14c in the *FGT Attribute Access Method Example* topic. • Reorganized information in the *Implementing the F-Tile Reference and System PLL Clocks Intel FPGA IP* chapter to make it clearer. • Removed note for *Enable FGT CDR Output #0* and *Enable FGT CDR Output #1* parameters in the *F-Tile Reference and System PLL Clocks Intel FPGA IP Parameters* table. • Added description to specify `qsf` location assignment for the *out_cdrclk_i* port in the *F-Tile Reference and System PLL Clocks Intel FPGA IP Port List* table. • Added new section *Guidelines for F-Tile Reference and System PLL Clocks Intel FPGA IP Usage*. • Updated step 5 of the *Hardware Flow Using the F-Tile Global Avalon Memory-Mapped Interface Intel FPGA IP* section. • Updated *F-tile PMA/FEC Direct PHY Design Implementation* chapter to remove references to design example. • Updated *F-Tile Transceiver Toolkit GUI*, *Collection View Tab of the F-Tile Transceiver Toolkit GUI*, *Toolkit Explorer*, *Example BER Test Setup and Results for the FGT PMA* figures in the *Debugging F-Tile Transceiver Links* chapter. • Updated PMA naming in the *Running BER Tests* section. • Updated *Transceiver Toolkit Parameter Settings* table with new information. • Updated *Creating Transceiver Links* section with *Import Collections* and *Export Collections* details. • Added footnote for *TX Equalization Parameters* in the *Transceiver Toolkit Parameter Settings* table.
2021.10.15	21.3	Made the following changes: • Updated the *Preserving Unused PMA Lanes* section. • Updated the *Number of system copies* parameter in *General and Common Datapath Options* table. • Added *Enable Core PLL mode* parameter in *TX FGT Datapath* table. • Updated *Enable FHT RX data profile* parameter in *RX FHT PMA Parameters* table. • Updated the *Example Design Generation* topic in the *Configuring the IP* section. • Updated parameter names to match with GUI names in *Avalon Memory Mapped Interface Parameters* table. • Added description for *Number of system copies* parameter in the *Signal and Port Reference* section. • Updated description of the *FGT PMA Fractional Mode* section. • Added new topic *Run-time Reset Sequence Approximate Time Duration* in *Run-time Reset Sequence—TX + RX* section. • Updated step 4 onwards in the *Run-time Reset Sequence—TX with FEC* section

continued...

Document Version	Intel Quartus Prime Version	Changes
		• Updated description of the *Lane Offset Address* section. • Added new topic *Logical Avalon Memory-Mapped Port Indexing* in *Configuration Registers* section. • Updated steps in *FGT Attribute Access Method Example*. • Added footnote that *ETHERNET_FREQ_805_322* is not supported in section *Mode of System PLL - System PLL Reference Clock and Output Frequencies*. • Added new section *Hardware Flow Using the F-Tile Global Avalon Memory-Mapped Interface Intel FPGA IP* in *Implementing the F-Tile Global Avalon Memory-Mapped Interface Intel FPGA IP* chapter. • Added the following new sections in the *F-tile PMA/FEC Direct PHY Design Example Implementation* chapter. — *Implementing a RS-FEC Direct Design in the F-Tile PMA/FEC Direct PHY Intel FPGA IP*. — *PAM4 Encoding Schemes in Simulation*. — *F-tile Interface Planner Design Example*. — Updated the *Simulating the F-tile PMA/FEC Direct PHY Design Example* section. • Added and updated the following sections in the *Supported Tools* chapter. — *F-Tile PMA and FEC Direct Port Mapping Calculator*. — *F-Tile Clocking and Datapath Tool*. — *F-Tile TX Equalizer Tool*. • Added new chapter *Debugging F-Tile Transceiver Links*.
2021.08.18	21.2	• Added the following new sections and updated table in *Implementing the F-Tile PMA/FEC Direct PHY Intel FPGA IP* chapter: — *Configuration Registers*. — *Configurable Intel Quartus Prime Software Settings*. — *Configuring the F-Tile PMA/FEC Direct PHY Intel FPGA IP for Hardware Testing*. — *Hardware Configuration Using the Avalon Memory-Mapped Interface*. — Added loopback mode in *TX FHT PMA Parameters* table. • Added new topic in *F-Tile Placement Rules* section in the *F-Tile Architecture* chapter: — *Preserving Unused PMA Lanes*. • Added new chapter *Supported Tools*. • Added new chapter *Document Revision History for F-tile Architecture and PMA and FEC Direct PHY IP User Guide*. — Consolidated the *Document Revision History* section of each chapter into this chapter.
2021.07.23	21.2	Updated `tx_am_gen_start` and `tx_am_gen_2x_ack` signal directions in the following tables: • *Reset Signals* table. • *Reset Signal Descriptions* table.
2021.06.24	21.2	Initial document release.

A. Appendix

A.1. F-Tile Production Revision and Firmware OPNs

OPNs with F-tile Production Revision and Firmware:

- Intel Agilex 7 F-tile ES devices:
  - AGIx040R39AxxxxR0
  - AGIx022R31BxxxxAA
  - AGIx027R31BxxxxAA
  - AGIx027R29AxxxxR3
  - AGIx023R18AxxxxR0
  - AGFx022R31CxxxxAA
  - AGFx027R31CxxxxAA
  - AGFx027R24CxxxxR2
  - AGFx012R24CxxxxAA
  - AGFx014R24CxxxxAA
  - AGMx032R47AxxxxR0
  - AGMx039R47AxxxxR0
- All Intel Agilex 7 F-tile production devices.

Related Information

Intel Agilex 7 FPGAs and SoCs Device OPN Options