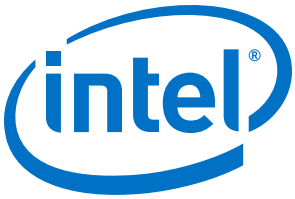


E-Tile Transceiver PHY User Guide

Subscribe
Send Feedback

UG-20056 | 2020.01.31
Latest document on the web: PDF | HTML

Contents

1. E-Tile Transceiver PHY Overview

The E-tile is a 24-channel, PAM4/NRZ dual-mode transceiver tile that is used in multiple variants of the Intel® Stratix® 10 and Intel Agilex™ device families. Refer to the respective *Product Tables* and *Pin-Out Files for Intel FPGA Devices* to find the actual number of transceivers available in each device.

Below is the performance comparison of E-tile to L-tile and H-tile transceiver tiles.

Table 1. Transceiver Tile Variants

Tile	Channel Type	Channel Capability		Channel Hard IP access
		Chip-to-Chip	Backplane	
L-tile	GX	17.4 Gbps (Non-Return-to-Zero (NRZ))	12.5 Gbps (NRZ)	PCIe* Gen3x16
	GXT	26.6 Gbps (NRZ)		
H-tile	GX	17.4 Gbps (NRZ)	17.4 Gbps (NRZ)	PCIe Gen3x16 50G/100G Ethernet MAC Firecode Forward Error Correction (FEC)
	GXT	28.3 Gbps (NRZ)	28.3 Gbps (NRZ)	
E-tile	GXE	28.9 Gbps (NRZ), 57.8 Gbps (pulse amplitude modulation (PAM4))		10G/25G/100G Ethernet MAC Reed Solomon Forward Error Correction (RS-FEC)

The transceiver tiles are connected to the FPGA fabric using Intel's Embedded Multi-die Interconnect Bridge (EMIB) technology.

Related Information

Pin-Out Files for Intel FPGA Devices

1.1. Supported Features

Table 2. Features Supported in E-Tile Transceivers

Feature	Description
Total transceivers	24 dual mode channels per tile • 24 channels available in NRZ mode for data rates from 1 Gbps to 28.9 Gbps • 24 channels available in PAM4 mode for data rates from 2 Gbps to 30 Gbps • 12 channels available in PAM4 mode for data rates from 30 Gbps to 57.8 Gbps
10G/25G/100G Ethernet with optional 1588 capability + RS-FEC (528, 514)/RS-FEC (544, 514)	Hard IP

1.2. E-Tile Layout in Intel Stratix 10 Device Variants

Intel Stratix 10 TX or MX FPGA configurations support E-tile transceivers.

Intel Stratix 10 MX device configurations combine FPGAs with high-bandwidth memory.

Intel Stratix 10 DX device configurations include both E-tile and P-tile transceivers.

1.2.1. Intel Stratix 10 TX H-Tile and E-Tile Configurations

Intel Stratix 10 TX FPGAs offer transceiver capability by combining H-tiles and E-tiles. This section lists all possible TX FPGA configurations.

Figure 1. Intel Stratix 10 TX Device with 1 E-Tile (24 Transceiver Channels)

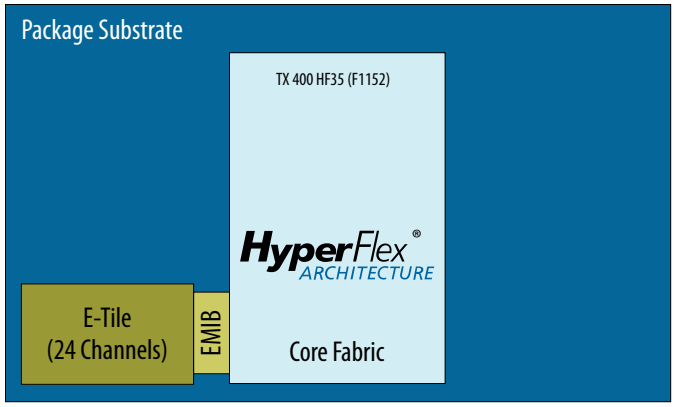


Figure 2. Intel Stratix 10 TX Device with 1 E-Tile and 1 H-Tile (48 Transceiver Channels)

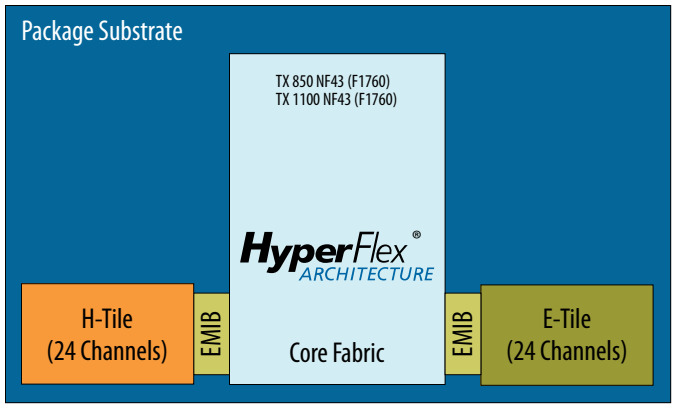


Figure 3. Intel Stratix 10 TX Device with 2 E-Tiles and 1 H-Tile (72 Transceiver Channels)

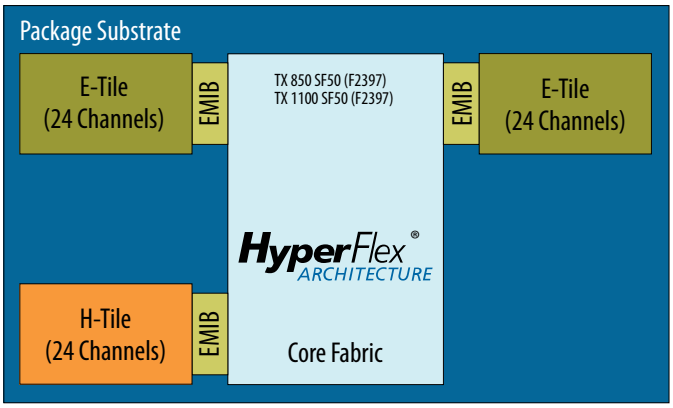


Figure 4. Intel Stratix 10 TX Device with 3 E-Tiles and 1 H-Tile (96 Transceiver Channels)

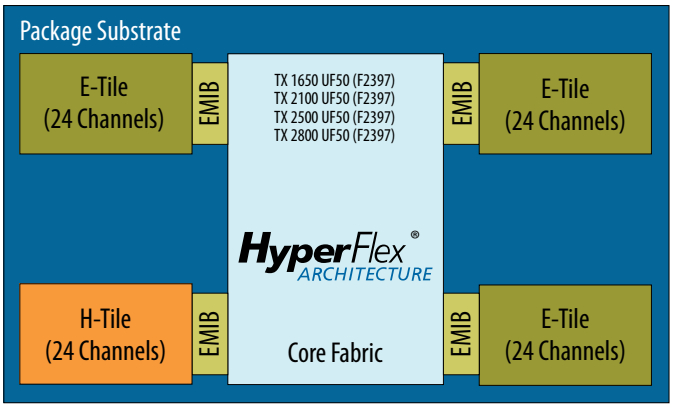
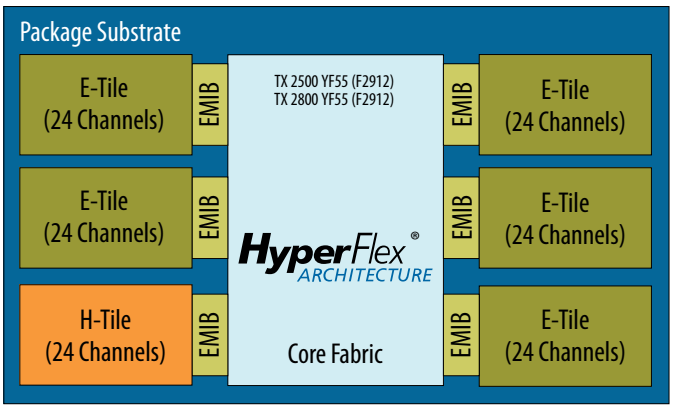


Figure 5. Intel Stratix 10 TX Device with 5 E-Tiles and 1 H-Tile (144 Transceiver Channels)

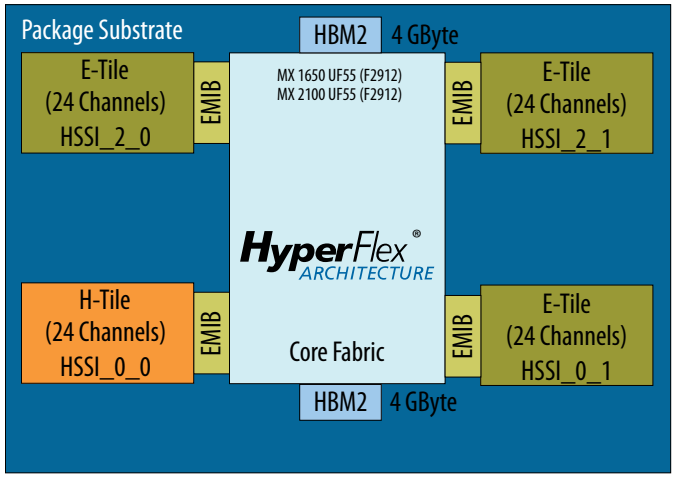


There is no package migration between Intel Stratix 10 GX/SX and Intel Stratix 10 TX device families (H-tile and E-tile).

1.2.2. Intel Stratix 10 MX H-Tile and E-Tile Configurations

Intel Stratix 10 MX devices combine the programmability and flexibility of Intel Stratix 10 FPGAs and SoCs with 3D stacked high-bandwidth memory 2 (HBM2). The dynamic random access memory (DRAM) tile is physically connected to the FPGA using Intel's Embedded Multi-die Interconnect Bridge (EMIB) technology.

Figure 6. Intel Stratix 10 MX Device with 3 E-Tiles, 1 H-Tile (96 Transceiver Channels) and 2 HBM2



There is no package migration between Intel Stratix 10 MX and Intel Stratix 10 TX device families (H-tile and E-tile) or Intel Stratix 10 GX/SX device families.

1.2.3. Intel Stratix 10 DX P-Tile and E-Tile Configurations

Intel Stratix 10 DX devices combine P-tiles for processor connectivity along with E-tiles for Ethernet support.

Table 3. Available E-Tile Transceiver Channels in Intel Stratix 10 DX FPGA Devices

Intel Stratix 10 DX Device Name	Number of E-Tile Transceiver Channels	Available E-Tile Transceiver Channel Locations
DX 1100	16	0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 14, 15, 20, 21, 22, 23
DX 2100	24	0 through 23
DX 2800	8	0, 1, 2, 3, 12, 13, 14, 15

Refer to the respective *Pin-Out Files for Intel FPGA Devices* to find the actual number of reference clocks available in each device.

Figure 7. Intel Stratix 10 DX Device with 1 P-Tile and 1 E-Tile (32 Transceiver Channels)

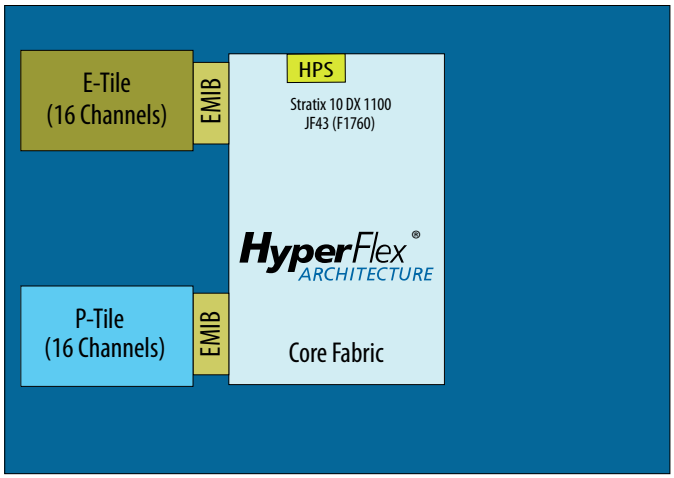


Figure 8. Intel Stratix 10 DX Device with 3 P-Tiles and 1 E-Tile (84 Transceiver Channels)

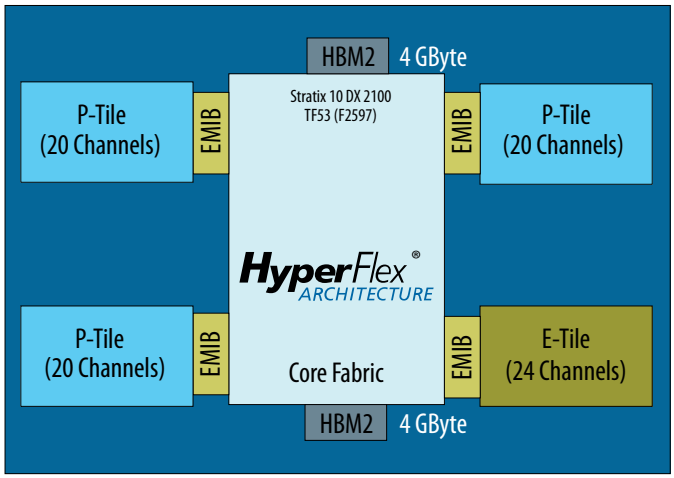
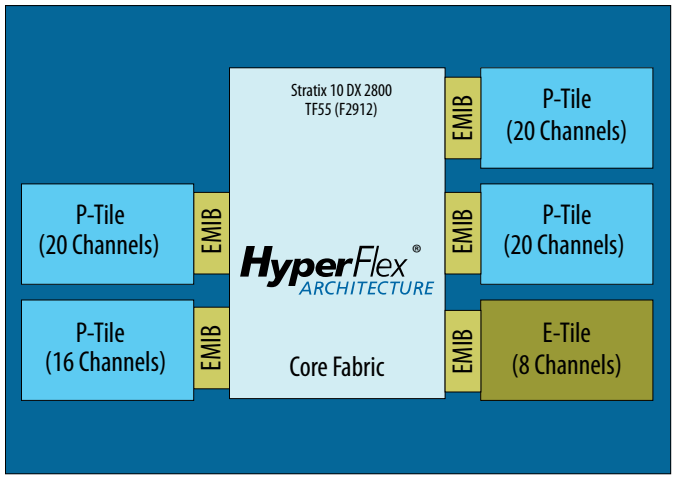


Figure 9. Intel Stratix 10 DX Device with 4 P-Tiles and 1 E-Tile (84 Transceiver Channels)



Related Information

Pin-Out Files for Intel FPGA Devices

1.3. E-Tile Layout in Intel Agilex F-Series Device Variants

The Intel Agilex F-Series device configuration includes both E-tile and P-tile transceivers.

Table 4. Available E-Tile Transceiver Channels in Intel Agilex FPGA Devices

Package	Number of E-Tile Transceiver Channels	Available E-Tile Transceiver Channel Locations
R2486A	16	0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 14, 15, 20, 21, 22, 23
R2013A R2514A	24	0 through 23

Refer to the respective *Pin-Out Files for Intel FPGA Devices* to find the actual number of reference clocks available in each device.

Figure 10. Intel Agilex Device with 1 P-Tile and 1 E-Tile (40 Transceiver Channels)

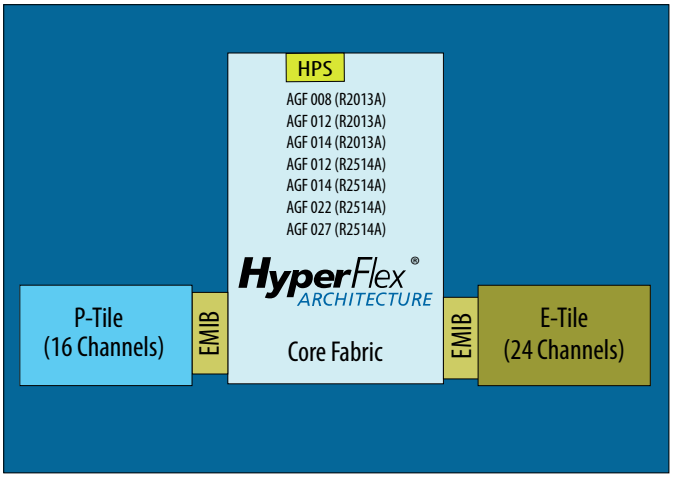
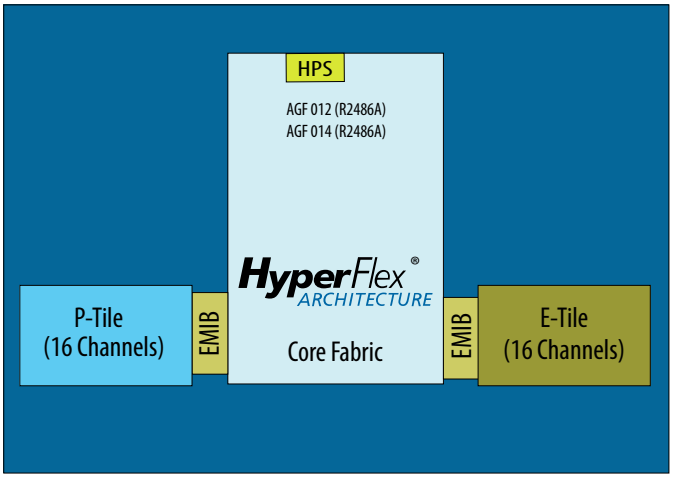


Figure 11. Intel Agilex Device with 1 P-Tile and 1 E-Tile (32 Transceiver Channels)



Related Information

Pin-Out Files for Intel FPGA Devices

1.4. Transceiver Counts

Table 5. Transceiver Counts in Intel Stratix 10 TX Devices with E-Tiles (NF43, SF50, UF50, YF55)

The number in the Intel Stratix 10 TX Device Name column indicates the device's Logic Element (LE) count (in thousands LEs).

Intel Stratix 10 TX Device Name	F1152 HF35 (35 mm x 35 mm) Transceivers (E, H)	F1760 NF43 (42.5 mm x 42.5 mm) Transceivers (E, H)	F2397 SF50, UF50 (50 mm x 50 mm) Transceivers (E, H)	F2912 YF55 (55 mm x 55 mm) Transceivers (E, H)
TX 400	24, 0	N/A	N/A	N/A
TX 850	N/A	24, 24	48, 24	N/A
TX 1100	N/A	24, 24	48, 24	N/A
TX 1650	N/A	N/A	72, 24	N/A
TX 2100	N/A	N/A	72, 24	N/A
TX 2500	N/A	N/A	72, 24	120, 24
TX 2800	N/A	N/A	72, 24	120, 24

Table 6. Transceiver Counts in Intel Stratix 10 MX Devices with E-Tiles (UF55)

The number in the Intel Stratix 10 MX Device Name column indicates the device's Logic Element (LE) count (in thousands LEs).

Intel Stratix 10 MX Device Name	F2912 UF55 (55 mm x 55 mm) Transceivers (E, H)
MX 1650	72, 24
MX 2100	72, 24

Table 7. Transceiver Counts in Intel Stratix 10 DX Devices with E-Tiles (JF43, TF53, TF55)

The number in the Intel Stratix 10 DX Device Name column indicates the device's Logic Element (LE) count (in thousands LEs).

Intel Stratix 10 DX Device Name	F1760 JF43 (42.5 mm x 42.5 mm) Transceivers (P, E)	F2597 TF53 (52.5 mm x 52.5 mm) Transceivers (P, E)	F2912 TF55 (55 mm x 55 mm) Transceivers (P, E)
DX 1100	16, 16	N/A	N/A
DX 2100	N/A	60, 24	N/A
DX 2800	N/A	N/A	76, 8

Table 8. Transceiver Counts in Intel Agilex Devices with E-Tile (R2013A, R2486A, R2514A)

The number in the Intel Agilex Device Name column indicates the device's Logic Element (LE) count (in thousands LEs).

Intel Agilex Device Name	R2013A (47 mm x 38 mm) Transceivers (P, E)	R2486A (55 mm x 42.5 mm) Transceivers (P, E)	R2514A (45 mm x 45 mm) Transceivers (P, E)
AGF 008	16, 24	N/A	N/A
AGF 012	16, 24	16, 16	16, 24
AGF 014	16, 24	16, 16	16, 24
AGF 022	N/A	N/A	16, 24
AGF 027	N/A	N/A	16, 24

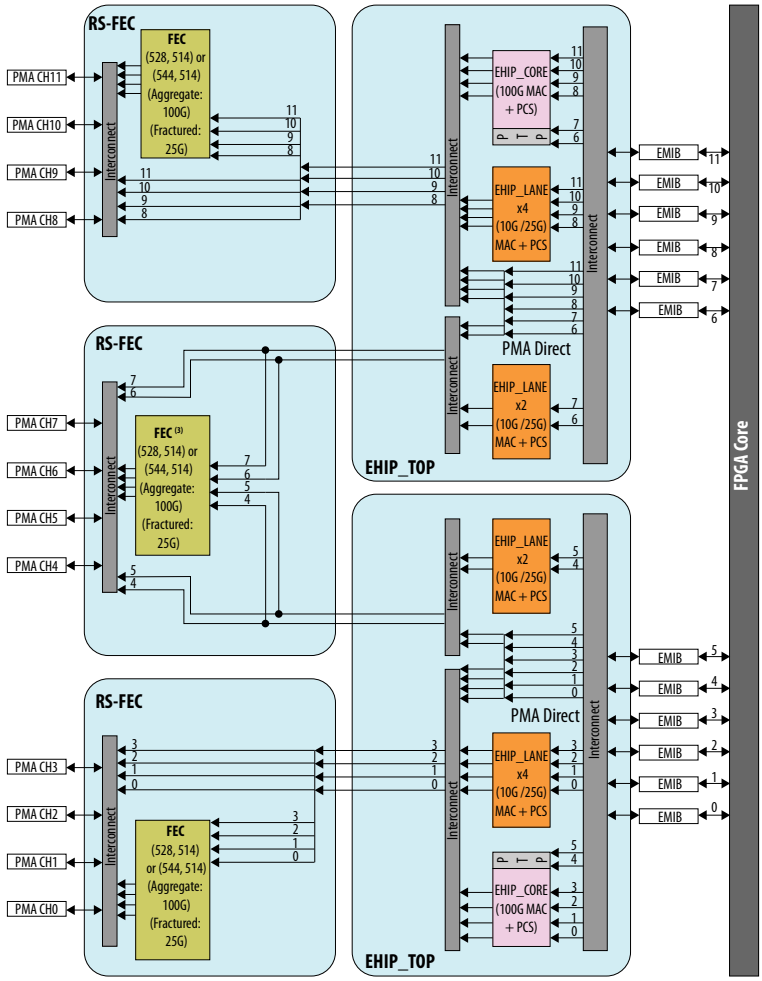
1.5. E-Tile Building Blocks

E-tile transceivers consist of the following individual blocks:

- Transceiver channels
- Reference clock network
- Ethernet Hard IP (EHIP_LANE / EHIP_CORE)
- Reed-Solomon Forward Error Correction (RS-FEC)
- 1588 PTP for Ethernet

Figure 12. E-Tile Architecture and Datapath Overview

Showing 12 out of 24 channels per tile.



Notes:

1. Not all datapath combinations are available.
2. Datapath enablement depends on the configuration you are implementing. Refer to the E-Tile Channel Placement tool for possible configurations.
3. This FEC block can only be used in aggregate mode with FEC direct application (e.g. 128GFC Fibre-Channel). This FEC block cannot be used in in aggregate mode with EHIP_CORE because there is no EHIP_CORE in this location.

1.5.1. GXE Transceiver Channel

The E-tile offers up to 24 full-duplex transceiver channels. These channels provide continuous data rates from 1 Gbps to 28.9 Gbps in NRZ mode, and 2 Gbps to 57.8 Gbps in PAM4 mode. For longer-reach backplane driving applications, adaptive equalization circuits are available to equalize the system losses.

The Physical Medium Attachment (PMA) provides interfacing capabilities to the following physical channels.

- Transmitter (TX)
- Receiver (RX)
- High speed clocking resources

Within a tile, you can configure channels as both bonded and non-bonded in duplex operation. Nine reference clock pins drive mux logic that allows you to direct reference clocks (`refclk_in_A` and `refclk_in_B`) to the transmitter/receiver to enable these features. For more details about the reference clocks, refer to the *Clock Network* chapter.

Related Information

Clock Network on page 112

1.5.2. GXE Channel Usage

Channel usage depends on your channel configuration. In NRZ mode, all 24 GXE channels in a tile are available. When the channel is configured in PAM4 mode and the data rate is greater than 30 Gbps, two adjacent core interfaces are combined to provide a single PAM4 channel.

Note: You can configure multiple data rates across different PAM4 channels < 30 Gbps simultaneously.

Figure 13. GXE Channel Usage Example: Channels Running at Data Rates > 30 Gbps PAM4 PMA Direct Mode without RS-FEC

12 even-numbered channels are available in a tile when the data rate is greater than 30 Gbps.

Intel® Stratix 10® E-tile Floor Plan					Channel Settings				
XCVR PMA	RSFEC	EHIP_TOP		Core Interface	Protocol	Level	Baud Rate	FEC mode	RS FEC
Channel_23	RSFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23	None	None	None	Off	Off
Channel_22			EHIPLANE_22	Interface_22	PMA direct	PAM4	2G-57.8G		Off
Channel_21			EHIPLANE_21	Interface_21	None	None	None		Off
Channel_20			EHIPLANE_20	Interface_20	PMA direct	PAM4	2G-57.8G		Off
Channel_19	RSFEC_4		EHIPLANE_19	Interface_19	None	None	None	Off	Off
Channel_18			EHIPLANE_18	Interface_18	PMA direct	PAM4	2G-57.8G		Off
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17	None	None	None		Off
Channel_16			EHIPLANE_16	Interface_16	PMA direct	PAM4	2G-57.8G		Off
Channel_15	RSFEC_3		EHIPLANE_15	Interface_15	None	None	None	Off	Off
Channel_14			EHIPLANE_14	Interface_14	PMA direct	PAM4	2G-57.8G		Off
Channel_13			EHIPLANE_13	Interface_13	None	None	None		Off
Channel_12			EHIPLANE_12	Interface_12	PMA direct	PAM4	2G-57.8G		Off
Channel_11	RSFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11	None	None	None	Off	Off
Channel_10			EHIPLANE_10	Interface_10	PMA direct	PAM4	2G-57.8G		Off
Channel_9			EHIPLANE_9	Interface_9	None	None	None		Off
Channel_8			EHIPLANE_8	Interface_8	PMA direct	PAM4	2G-57.8G		Off
Channel_7	RSFEC_1		EHIPLANE_7	Interface_7	None	None	None	Off	Off
Channel_6			EHIPLANE_6	Interface_6	PMA direct	PAM4	2G-57.8G		Off
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5	None	None	None		Off
Channel_4			EHIPLANE_4	Interface_4	PMA direct	PAM4	2G-57.8G		Off
Channel_3	RSFEC_0		EHIPLANE_3	Interface_3	None	None	None	Off	Off
Channel_2			EHIPLANE_2	Interface_2	PMA direct	PAM4	2G-57.8G		Off
Channel_1			EHIPLANE_1	Interface_1	None	None	None		Off
Channel_0			EHIPLANE_0	Interface_0	PMA direct	PAM4	2G-57.8G		Off

Figure 14. GXE Channel Usage Example: Channels Running at Data Rates < 30 Gbps PAM4/NRZ PMA Direct Mode without RS-FEC

Intel® Stratix 10® E-tile Floor Plan					Channel Settings				
XCVR PMA	RSFEC	EHIP_TOP		Core Interface	Protocol	Level	Rate	FEC mode	RSFEC
Channel_23	RSFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23	PMA direct	PAM4low	2G-30G	Off	Off
Channel_22			EHIPLANE_22	Interface_22	PMA direct	PAM4low	2G-30G		Off
Channel_21			EHIPLANE_21	Interface_21	PMA direct	PAM4low	2G-30G		Off
Channel_20			EHIPLANE_20	Interface_20	PMA direct	PAM4low	2G-30G		Off
Channel_19	RSFEC_4		EHIPLANE_19	Interface_19	PMA direct	PAM4low	2G-30G	Off	Off
Channel_18			EHIPLANE_18	Interface_18	PMA direct	PAM4low	2G-30G		Off
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17	PMA direct	PAM4low	2G-30G		Off
Channel_16			EHIPLANE_16	Interface_16	PMA direct	PAM4low	2G-30G		Off
Channel_15	RSFEC_3		EHIPLANE_15	Interface_15	PMA direct	PAM4low	2G-30G	Off	Off
Channel_14			EHIPLANE_14	Interface_14	PMA direct	PAM4low	2G-30G		Off
Channel_13			EHIPLANE_13	Interface_13	PMA direct	PAM4low	2G-30G		Off
Channel_12			EHIPLANE_12	Interface_12	PMA direct	PAM4low	2G-30G		Off
Channel_11	RSFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11	PMA direct	NRZ	1G-28.9G	Off	Off
Channel_10			EHIPLANE_10	Interface_10	PMA direct	NRZ	1G-28.9G		Off
Channel_9			EHIPLANE_9	Interface_9	PMA direct	NRZ	1G-28.9G		Off
Channel_8			EHIPLANE_8	Interface_8	PMA direct	NRZ	1G-28.9G		Off
Channel_7	RSFEC_1		EHIPLANE_7	Interface_7	PMA direct	NRZ	1G-28.9G	Off	Off
Channel_6			EHIPLANE_6	Interface_6	PMA direct	NRZ	1G-28.9G		Off
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5	PMA direct	NRZ	1G-28.9G		Off
Channel_4			EHIPLANE_4	Interface_4	PMA direct	NRZ	1G-28.9G		Off
Channel_3	RSFEC_0		EHIPLANE_3	Interface_3	PMA direct	NRZ	1G-28.9G	Off	Off
Channel_2			EHIPLANE_2	Interface_2	PMA direct	NRZ	1G-28.9G		Off
Channel_1			EHIPLANE_1	Interface_1	PMA direct	NRZ	1G-28.9G		Off
Channel_0			EHIPLANE_0	Interface_0	PMA direct	NRZ	1G-28.9G		Off

The figure above shows an example of an E-tile in a device package that supports all 24 transceiver channels. Based on product family and device package, the number of transceivers in an E-tile can vary. For the exact number of available transceiver channels, refer to the *Pin-Out Files for Intel FPGA Devices*. Related information can also be found in the *E-tile Channel Placement Tool*.

Related Information

- Pin-Out Files for Intel FPGA Devices
- E-tile Channel Placement Tool

1.5.3. Reference Clocks

E-tile transceivers include a reference clock network for clocking flexibility and channel bonding. There are up to nine low-voltage positive/pseudo emitter-coupled logic (LVPECL[1]) reference clock pins on the tile, which are dynamically selectable through two inputs, `refclk_in_A` and `refclk_in_B`, to drive the transmitter/receiver. You can configure the pins as either 2.5-V LVPECL compliant or 3.3-V LVPECL tolerant.

(1) Refer to the *Device Data Sheet* for the acceptable LVPECL specifications.

Intel recommends that you use the default setting, which includes source termination at 2.5 V and AC coupling caps. The *Device Data Sheet* provides the electrical characteristics under the E-tile section. Additional important electrical information is available in the *Device Family Pin Connection Guidelines*.

Table 9. Key Reference Clock Considerations

Consideration	Description
Power	The reference clock pins support only the low-voltage positive/pseudo emitter-coupled logic (LVPECL) standard. The pins are internally terminated to 2.5 V by default, but are tolerant to 3.3 V as well. You can disable the termination and place external termination to either 2.5 V or 3.3 V. DC block caps and biasing resistors are fixed internally by default. The only requirement is that you meet the LVPECL specifications. The *Device Data Sheet* provides additional electrical characteristics under the E-tile section. The *Device Family Pin Connection Guidelines* also contains additional electrical characteristics .
Sharing	The nine reference clocks are shared, and they span across all 24 channels within a given E-tile. You must design the transceiver interface accordingly when you use the same reference clock source across multiple E-tiles. Reference clocks are not shared between transceiver tiles because there are no connections between transceiver tiles.
Inputs	Each of the 24 channels has a `refclk_in_A` input that receives one of the nine reference clocks, `refclk[8:0]`. The first, `refclk[0]`, is a low-skew balanced clock, and the other eight are non-skew balanced clocks. Only `refclk[0]` supports channel bonding, which is used mainly for TX clocking. When an RX channel is adjacent to a TX channel and is running at the same rate, you can share any of the reference clocks between the two channels. Each of the 24 channels also has a `refclk_in_B` input that only receives `refclk[1]`.
Rate switching	Use `refclk[1]` for rate switching. Use `refclk[1]` for rate switching or for different TX/RX reference clock frequencies.

The following figures demonstrate the usage of these nine reference clocks through the two inputs.

Figure 15. Dynamically Selected Reference Clocks

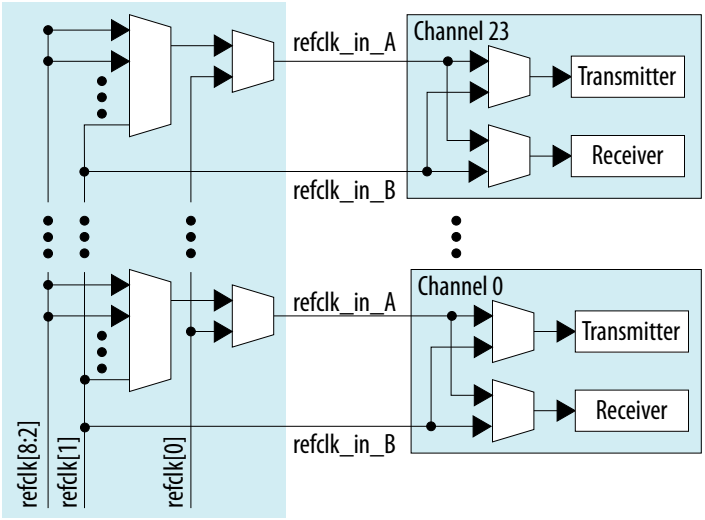
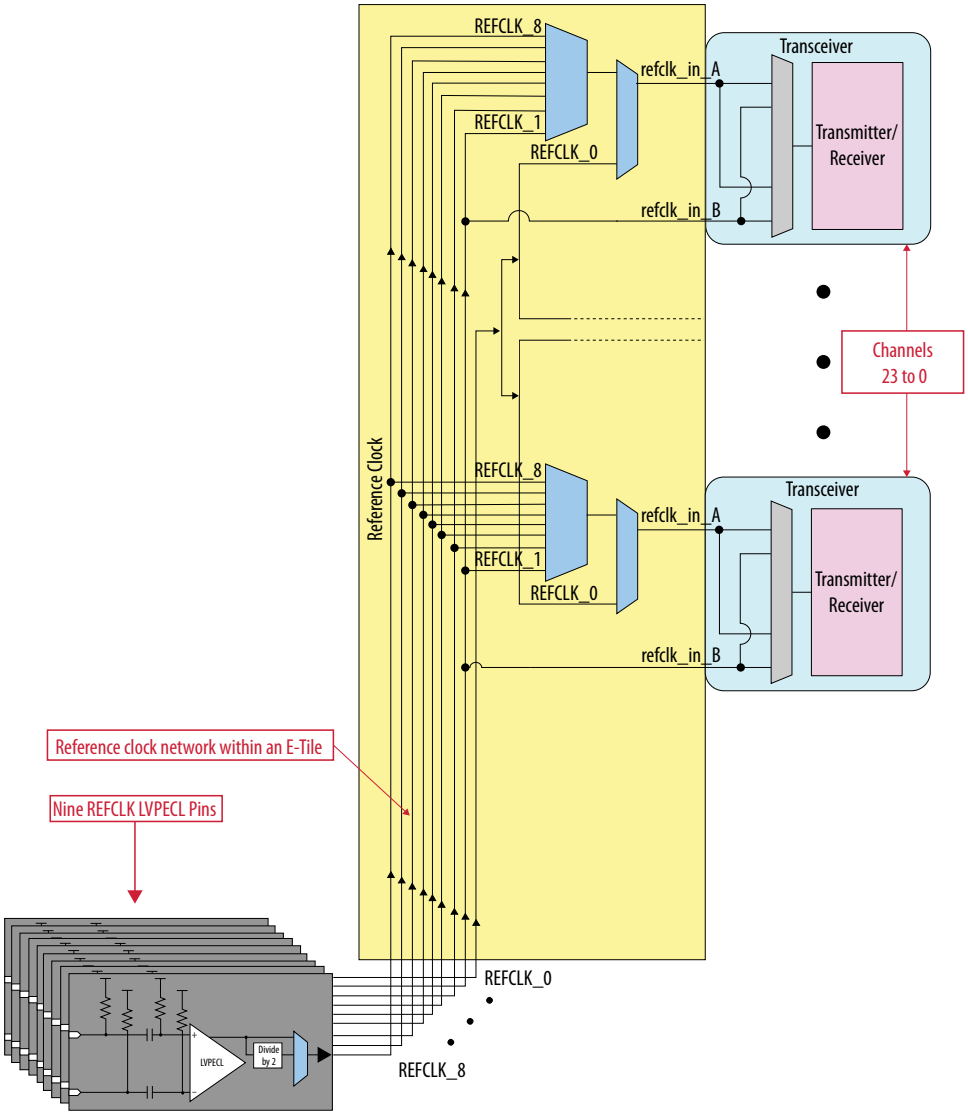
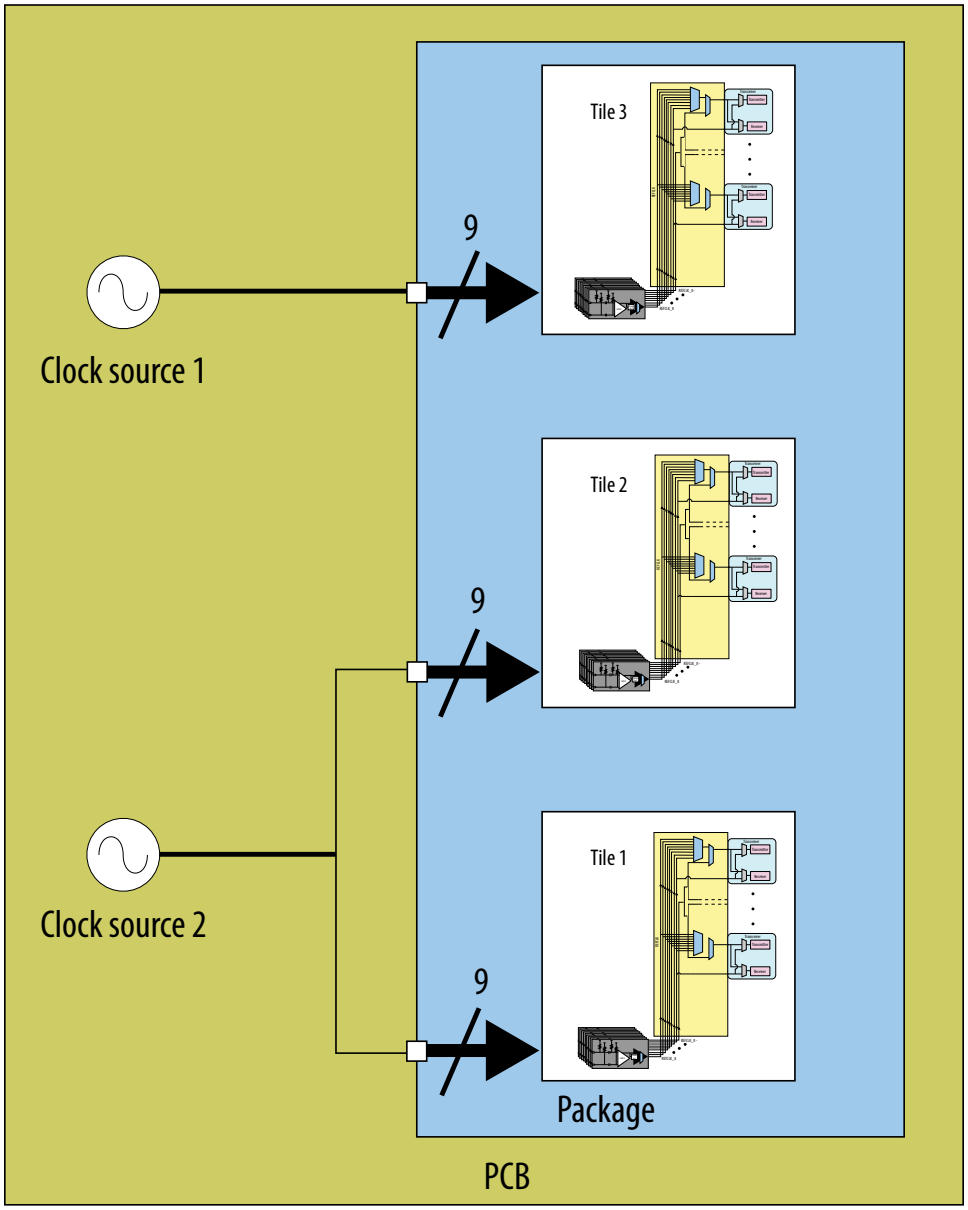


Figure 16. Reference Clock Access



The reference clock network spans across the entire transceiver tile. If the design requires a single reference clock to be supplied to more than one transceiver tile, you must route the reference clock to multiple tiles on the printed circuit board (PCB).

Figure 17. Single Reference Clock Used Across two E-Tiles



All 24 channels have access to all nine reference clock options. This provides:

- Full flexibility on selecting reference clocks on a per-channel basis
- Channel bonding enabled using `refclk[0]`

In full duplex mode, each channel can dynamically select any of the nine reference clocks. If RX and TX channels require different clock frequencies, `refclk[1]` must be used as one of the two clocks. Different TX and RX `refclk` is supported in simplex mode only; dual simplex is not supported.

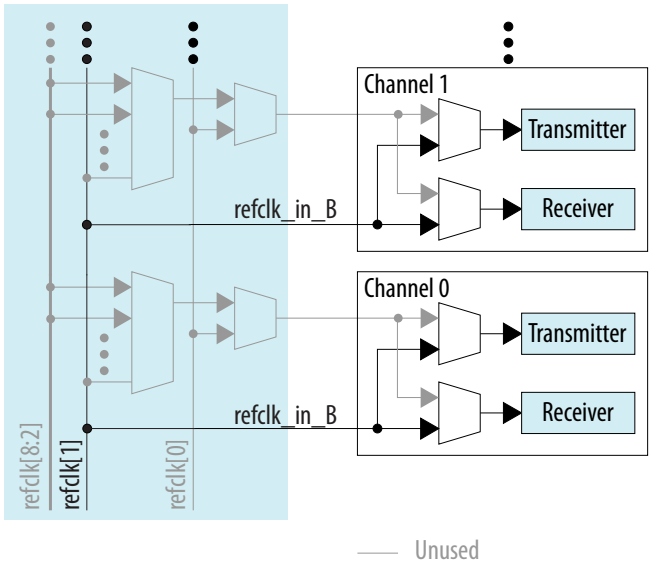
Channel bonding is a common technique used to minimize high speed serial lane-lane transmit skew for multi-lane protocols. Channel bonding is supported under the following conditions:

- Using `refclk[0]`
- NRZ PMA direct mode
- Data rate limited to:
  - 16-bit (parallel data width): 12.0 Gbps
  - 20-bit (parallel data width): 16.0 Gbps
  - 32-bit (parallel data width): 28.0 Gbps

Note: Bonding is only supported within a tile. Bonding is not supported within a package across different tiles, even if the reference clock is shared.

Figure 18. TX and RX with the Same Reference Clock

This configuration shows `refclk[1]` being used for TX and RX on both channels, enabling use of the same reference clock.



Related Information

- Intel Stratix 10 Device Data Sheet
- Intel Stratix 10 Device Family Pin Connection Guidelines
- Intel Agilex Device Data Sheet
- Intel Agilex Device Family Pin Connection Guidelines

1.5.4. Ethernet Hard IP (EHIP)

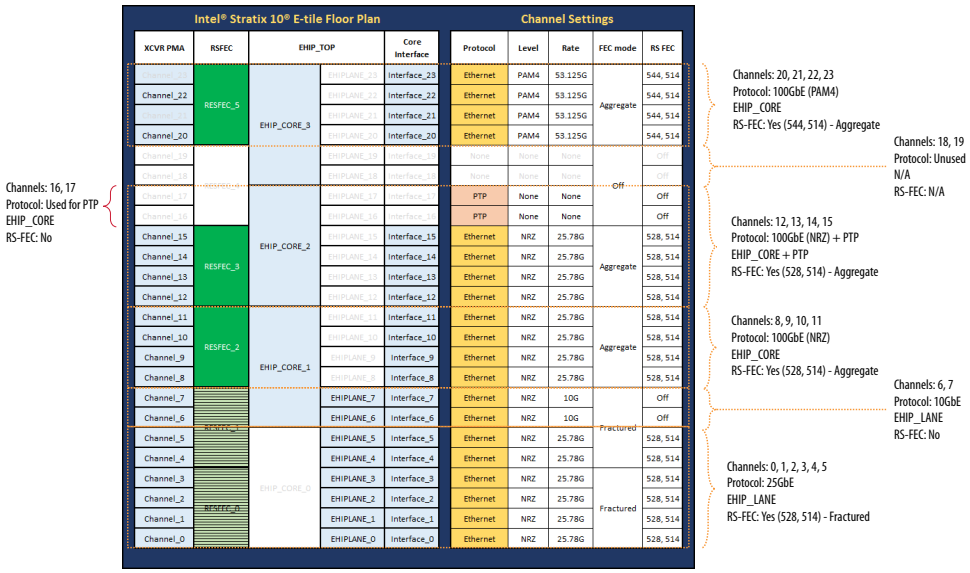
The Ethernet Hard IP is a hardened core of assorted multi-lane and single-lane Ethernet components.

E-tiles include four instances of the Ethernet Hard IP, which in turn supports up to four multi-lane Ethernet MAC stacks, or 24 channels of single-lane Ethernet channel (MAC/PCS) support.

Each Hard IP instance contains a full-featured multi-lane Ethernet (EHIP_CORE) Media Access Control (MAC) layer, which offers a number of interfacing options from the FPGA fabric. The multi-lane core can be used for 100G Ethernet applications. In addition to the multi-lane MAC stack, the Ethernet Hard IP contains six instances of a single-lane Ethernet channel.

Figure 19. Ethernet Hard IP Overview

Reed Solomon Forward Error Correction (RS-FEC) is configurable for single-lane 10 GbE or 25 GbE interfaces as well as multi-lane 100 GbE.



The E-tile implementation of the Ethernet Hard IP provides the following features and support:

- 4x hardened MACs per E-tile
- Each MAC block can be configured as:
  - One 100 GbE interface
  - Six 10 GbE / 25 GbE interfaces
  - Bypassable
- Supports IEEE 1588-2002 standard/Precision Time Protocol (PTP)
  - When used with the multi-lane 100 GbE core or 1-4 lanes of the 10 GbE or 25 GbE stack, two additional transceiver channels are configured for 1588. The location of these two additional channels is hardened for 1588 configuration. Use the *E-Tile Channel Placement Tool* to see how the channels are configured to support 1588.

Related Information

E-Tile Channel Placement Tool

1.5.5. Supported Applications/Modes

Table 10. Supported Applications/Modes

Supported Application/Mode	EHIP_CORE	EHIP_LANE	PMA Direct	RS-FEC	Dual Mode (2)	PAM4/NRZ
100GbE (4 x 25G) – NRZ w/ FEC	Yes	No	No	Yes (Aggregate)	No	NRZ
100GbE (2 x 50G) – PAM4 w/ FEC	Yes	No	No	Yes (Aggregate)	Yes	PAM4
100GbE (4 x 25G) – NRZ w/o FEC	yes	No	No	No	No	NRZ
128G Fibre Channel w/ FEC	No	No	Yes	Yes (Aggregate)	No	NRZ
32G Fibre Channel w/ FEC	No	No	Yes	Yes (Fractured)	No	NRZ
25GbE - NRZ w/FEC	No	Yes	No	Yes (Fractured)	No	NRZ
25GbE - NRZ w/o FEC	No	Yes	No	No	No	NRZ
CPRI 24G - NRZ w/ FEC	No	Yes	No	Yes (Fractured)	No	NRZ
10GbE – NRZ	No	Yes	No	No	No	NRZ
NRZ - 1G to 28.9G	No	No	Yes	No	No	NRZ
PAM4 - 2G to 30G	No	No	Yes	No	No	PAM4
PAM4 - 30G to 57.8G	No	No	Yes	No	Yes	PAM4
NRZ - 1G to 28.9G w/ FEC	No	No	Yes	Yes (Fractured)	No	NRZ

Note: Refer to the "E-Tile Transmitter and Receiver Data Rate Performance Specifications" table in the *Device Data Sheet* for speed grade information.

Related Information

- Intel Stratix 10 Device Data Sheet
- Intel Agilex Device Data Sheet

1.5.6. Feature Comparison Between Transceiver Tiles

Table 11. Transceiver Tile Feature Comparison

Feature	L-Tile/H-Tile	E-Tile
Native PHY IP	Configure NRZ mode	Configure NRZ and PAM4
PLL IP	ATXPLL, fPLL and CMU PLL IPs (available in the IP catalog)	Embedded in Native PHY and Ethernet Hard IPs
Reset controller IP	Reset controller IP (available in the IP catalog)	Embedded in the Native PHY and Ethernet Hard IP cores

continued...

(2) PMA Direct high data rate mode

Feature	L-Tile/H-Tile	E-Tile
Clocking modes	• TX PMA bonding up to 24 channels • ATXPLL-fPLL and fPLL-fPLL cascade • VCXO replacement (ATXPLL and fPLL fractional division support)	Only TX PMA bonding supported
Transceiver Calibration	Power-up calibration and recalibration	Power-up calibration and recalibration
Configuration ports	For each instantiated IP (Native PHY IP core, ATXPLL/fPLL), there is one configuration port.	For each instantiated Native PHY IP core, there are two configuration ports: one for the Native PHY IP core and another for RS-FEC.
Reconfiguration and register map	Registers available to configure the following: • PMA • PCS • ATXPLL • fPLL	Separate register map for the following: • PMA • RS-FEC • EHIP_LANE + EHIP_CORE • 1588 PTP
PCS	Available within the Native PHY IP core	Available within Ethernet Hard IP, not in the Native PHY IP core
Transmitter PMA	One post-tap and one pre-tap emphasis	One post-tap and three pre-tap emphasis for PAM4 One post-tap and one pre-tap emphasis for NRZ
Receiver PMA	Four RX adaptation modes: • Manual VGA, Manual CTLE, DFE Off • Adaptive VGA, Adaptive CTLE, DFE Off • Adaptive VGA, Adaptive CTLE, 1-Tap Adaptive DFE • Adaptive VGA, Adaptive CTLE, All-Tap Adaptive DFE	Two RX adaptation modes: • Continuous Adaptation • Initial Adaptation
Loopback paths	Serial, Pre-CDR Reverse Serial, Post-CDR Reverse Serial	Internal serial loopback (serial TX to serial RX) Reverse parallel loopback (parallel RX to parallel TX)
Hard PRBS	Available	Available
Hard PRBS error injection	Not available	Available
Eye viewer	On-Die Instrumentation through Transceiver Toolkit and Avalon®-MM (AVMM) access	Eye viewer available only through Transceiver Toolkit

1.6. E-Tile Transceiver PHY Overview Revision History

Document Version	Changes
2019.10.11	Made the following changes: • Re-wrote the overview to be E-tile-specific rather than product-specific. • Updated *Intel Stratix 10 TX H-Tile and E-Tile Configurations*. • Added Intel Stratix 10 DX device details in *E-Tile Layout in Intel Stratix 10 Device Variants*. • Added *Intel Stratix 10 DX P-Tile and E-Tile Configurations*. • Added the "Transceiver Counts in Intel Stratix 10 DX Devices with E-Tiles (JF43, TF53, TF55)" table. • Added *E-Tile Layout in Intel Agilex F-Series Device Variants*.

continued...

Document Version	Changes
	• Added the "Transceiver Counts in Intel Agilex Devices with E-Tiles (R2013A, R2486A, R2514A)" table. • Added the Related Information links for the Intel Agilex device documents. • Clarified that not all device packages support all 24 transceiver channels in an E-tile.
2019.07.29	Made the following change: • Changed *Intel Stratix 10 TX Advance Information Brief* to *Intel Stratix 10 TX Device Overview*.
2019.04.19	Made the following changes: • Added the "1588 PTP for Ethernet" building block. • Added "Different TX and RX `refclk` is supported in simplex mode only; dual simplex is not supported." • Added support for the 128G Fibre Channel w/ FEC mode.
2019.03.07	Made the following change: • Changed the data rate for E-tile Non-Return to Zero (NRZ) to 28.9 Gbps.
2019.02.04	Made the following changes: • Updated figures in "Intel Stratix 10 TX H-Tile and E-Tile Configurations". • Updated "Transceiver Counts in Intel Stratix 10 TX Devices with E-Tiles (NF43, SF50, UF50, YF55)." • Updated "Transceiver Counts in Intel Stratix 10 MX Devices with E-Tiles (UF55)."
2018.10.08	Made the following changes: • Changed the description of Inputs in the "Key Reference Clock Considerations" table. • Removed the "TX and RX with Different Reference Clocks" figure. • Clarified the descriptions for the transmitter PMA and loopback paths in the "Transceiver Tile Feature Comparison" table.
2018.07.18	Made the following changes: • Changed the PAM4 data rate to 57.8 Gbps in the "Transceiver Tile Variants" table.
2018.05.15	Made the following changes: • Updated the "GXE Channel Usage Example: Channels Running at Data Rates > 30 Gbps PAM4PMA Direct Mode without RS-FEC" figure • Updated the "GXE Channel Usage Example: Channels Running at Data Rates < 30 GbpsPAM4/NRZ PMA Direct Mode without RS-FEC" figure. • Updated the "Ethernet Hard IP Overview" figure. • Added the "Supported Applications/Modes" section. • Removed the 50GbE PAM4 w/ FEC application from the "Supported Applications/Modes" table.
2018.01.31	Initial release.

2. Implementing the Transceiver PHY Layer

2.1. Transceiver Design Flow in the Native PHY IP Core

The E-tile Native PHY IP core is the primary access point allowing you to access and customize the Native PHY IP core.

The E-tile Native PHY IP core supports the following usage modes:

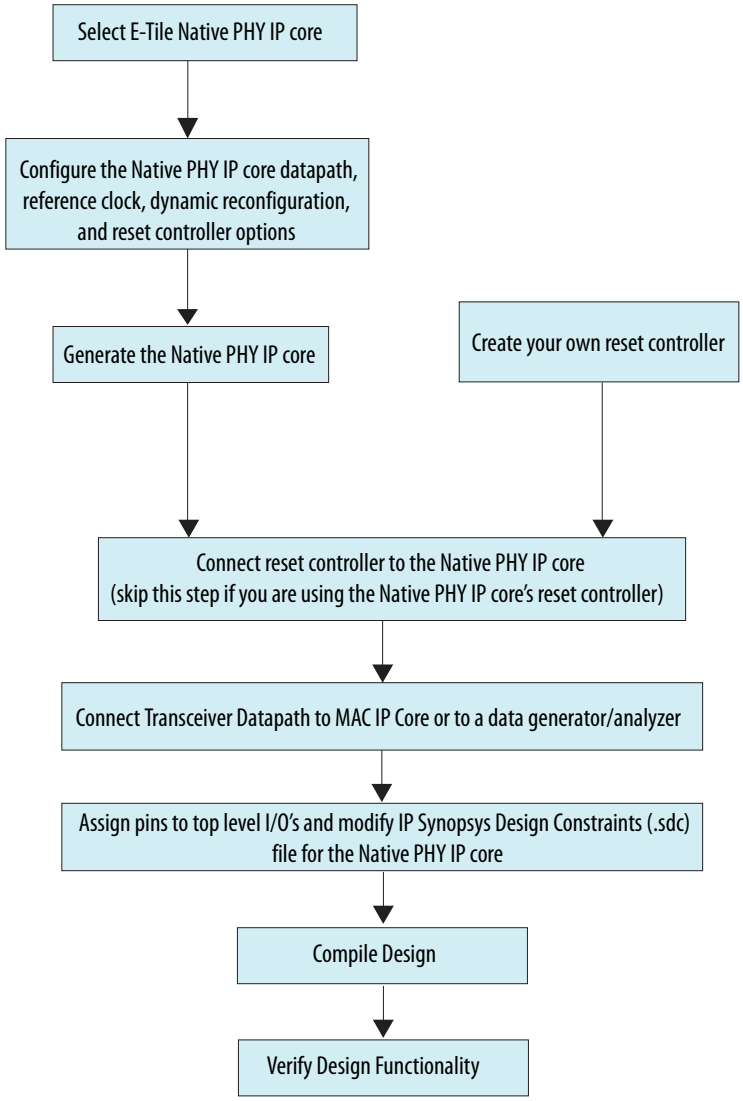
- PMA Direct
- PMA Direct high data rate PAM4
- Gearbox 64/66
- PLL

The PMA Direct usage mode is for PMA NRZ and PAM4 usage in the E-tile Native PHY IP core. The data is transferred directly between PMA interface and FPGA fabric through the EMIB. You can place a total of 24 PMA Direct channels in one E-tile. This mode is supported for both NRZ and PAM4 with the following PMA interface widths:

- 16
- 20
- 32
- 40
- 64 (Only in PMA Direct high data rate PAM4 mode)

The E-tile Native PHY IP core's integrated reset controller provides reset signals for the PMA Direct and PMA Direct high data rate PAM4 modes.

Figure 20. Transceiver Design Flow



2.1.1. E-Tile Native PHY IP Core

Much like the L- and H-Tile Native PHY IP Core, you have multiple options when instantiating the IP:

- Instantiating the Native PHY IP to interface to your own IP
- Not instantiating the Native PHY IP as apart of your own IP, and instead, providing a design example which contains both the MAC IP and the Native PHY IP instances

2.2. Configuring the Native PHY IP Core

The E-tile transceiver Native PHY IP core is the primary design entry tool, and provides direct access to E-tile transceiver PHY features.

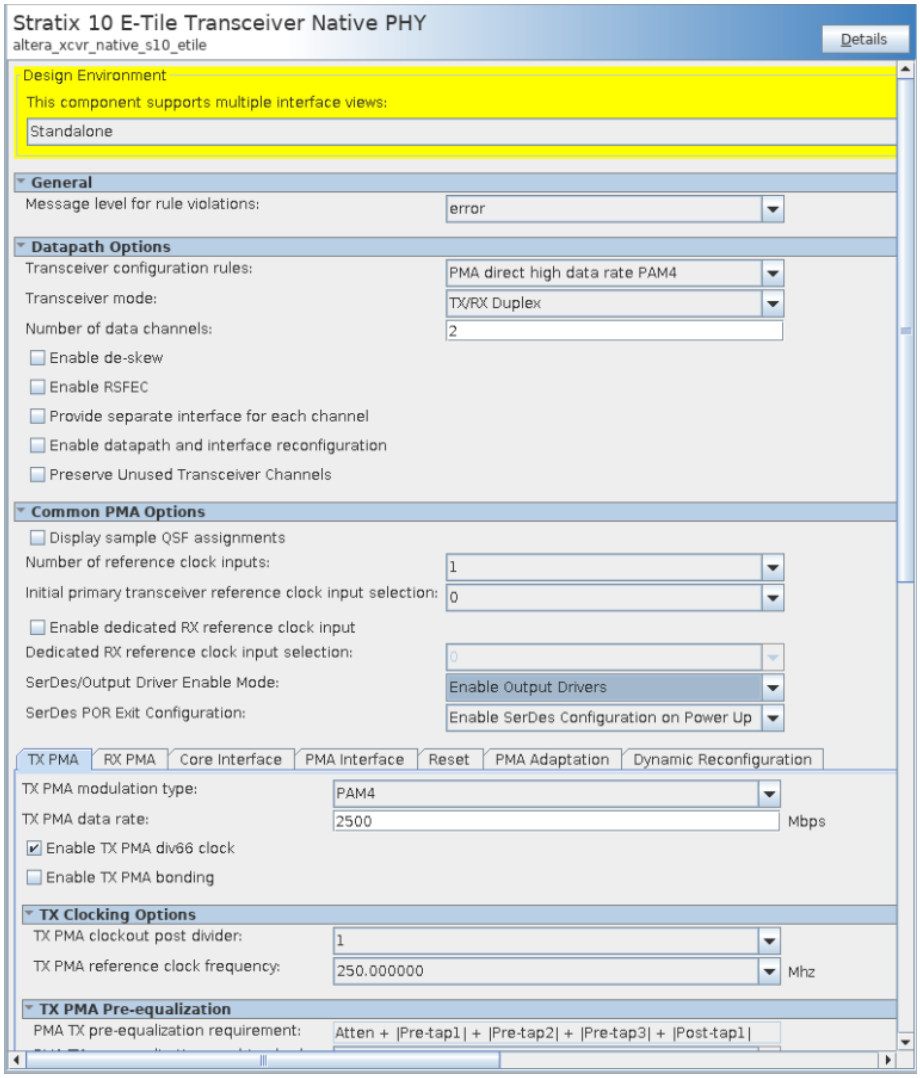
Use the Native PHY IP core in the Intel Quartus® Prime Pro Edition software to configure the transceiver PHY for your protocol implementation. To instantiate the IP:

1. Select the device family.
2. Click **Tools > IP Catalog** to select the E-tile transceiver Native PHY IP core.
3. Specify the IP parameters and configure the Native PHY IP core for your protocol implementation using the **Parameter Editor**.
4. Use the Native PHY IP core to instantiate one of the following transceiver usage modes:
   - PMA Direct
   - PMA Direct high data rate PAM4

   Based on the transceiver configuration rule that you select, the Native PHY IP core guides you to configure the transceiver appropriately.
5. After you configure the Native PHY IP core in the **Parameter Editor**, click **Generate HDL** to generate the IP instance.

   The top-level file generated with the IP instance includes all the available ports for your configuration. Use these ports to connect the Native PHY IP core to the clock network, the reset controller if you are not using Native PHY IP core's reset controller, and to other IP cores in your design.

Figure 21. Native PHY IP Core Parameter Editor



Note: Although the Intel Quartus Prime Pro Edition software provides legality checks, the supported FPGA fabric to transceiver interface widths and the supported data rates are pending characterization.

2.2.1. General and Datapath Parameters

You can customize your instance of the Native PHY IP core by specifying parameter values.

In the **Parameter Editor**, the parameters are organized in the following sections for each functional block and feature:

- General, Datapath Options, and Common PMA Options
- TX PMA
- RX PMA

- Core Interface
- PMA Interface
- Reset
- RS-FEC (when you enable this feature)
- Dynamic Reconfiguration

Figure 22. General, Datapath, and Common PMA Options

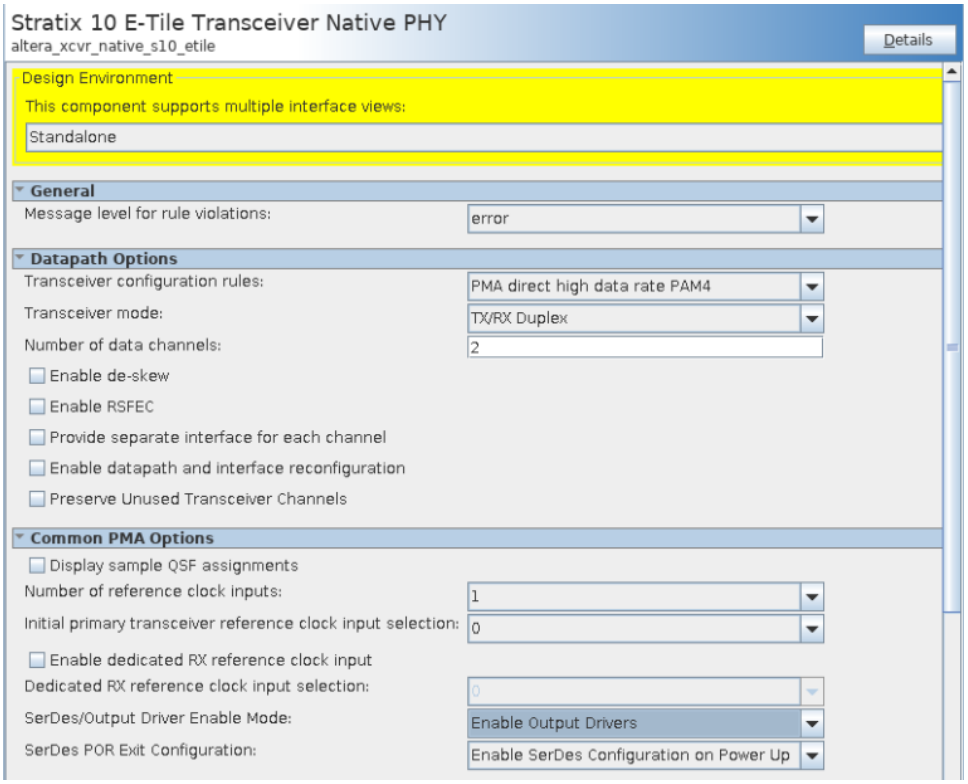


Table 12. General, Datapath Options, and Common PMA Options

Parameter	Value	Description
Message level for rule violations	**error** **warning**	Specifies the messaging level to use for parameter rule violations. Selecting **error** causes all rule violations to prevent IP generation. Selecting **warning** displays all rule violations as warnings in the message window and allows IP generation despite the violations.
Transceiver configuration rules	**PMA direct** **PMA direct high data rate PAM4** **Gearbox 64/66** **PLL**	Selects the protocol configuration rules for the transceiver. This parameter governs the rules for the correct settings of individual parameters. Certain features of the transceiver are available only for specific protocol configuration rules. This parameter is not a "preset". You must correctly set all other parameters for your specific protocol and application needs.
Transceiver mode	**TX/RX Duplex**	Specifies the operational mode of the transceiver: • TX/RX Duplex: Specifies a single channel that supports both transmission and reception. The default is TX/RX Duplex.

continued...

Parameter	Value	Description
Number of data channels	**1-24**	Specifies the number of transceiver channels you want to implement. The default value is 1. The maximum value is 24.
Enable de-skew	**On/Off**	Enables deskew for PAM4 high data rate (PMA direct mode only)
Enable RSFEC	**On/Off**	Enables the RS-FEC functionality.(3)
Provide separate interface for each channel	**On/Off**	When selected, the Native PHY IP core presents separate data, reset, and clock interfaces for each channel rather than a wide bus.
Enable datapath and interface reconfiguration	**On/Off**	Enables the ability to preconfigure and dynamically switch between the RS-FEC enabled and disabled modes.
Preserve Unused Transceiver Channels	**On/Off**	Preserves unused transceiver channels for PAM4 high data rate (PAM4 mode only)
Number of reference clock inputs	**1-5**	Specifies the desired number of reference clocks intended for the transmitter AND/OR receiver. This allows for dynamic clock source switching. Native PHY IP Core allows up to five clock inputs out of the possible nine for dynamic clock switching.
Initial TX reference clock input selection	**0**	This indicates the starting clock input selection used for this configuration when dynamically switching between multiple clock inputs.
Enable dedicated RX reference clock input	**On/Off**	Option to assign dedicated reference clock for the receiver instead of sharing it with the transmitter.
Dedicated RX clock input selection	**0-4**	When you enable the **Enable Receiver dedicated reference clock input** option, you can select the input clock with this parameter.
SerDes/Output Driver Enable Mode	**Enable Output Drivers** **Disable Output Drivers** **Disable PMA**	Specifies the operation mode of the serializer and deserializer and the output driver for active channels.
SerDes POR Exit Configuration	**Enable SerDes Configuration on Power Up** **Disable SerDes Configuration on Power Up**	Enables or disables the serializer and deserializer configuration on power-up. Normally, this should be enabled.

2.2.2. PMA Parameters

You can specify values for the following types of PMA parameters:

TX PMA:

- TX PMA Options
- TX PMA Pre-equalization
- TX Clocking Options

(3) When you enable RS-FEC, the number of data channels is 1-4.

RX PMA:

- RX PMA Options
- RX PMA Optional Ports
- RX Clocking Options

2.2.2.1. TX PMA Options

Figure 23. TX PMA Options

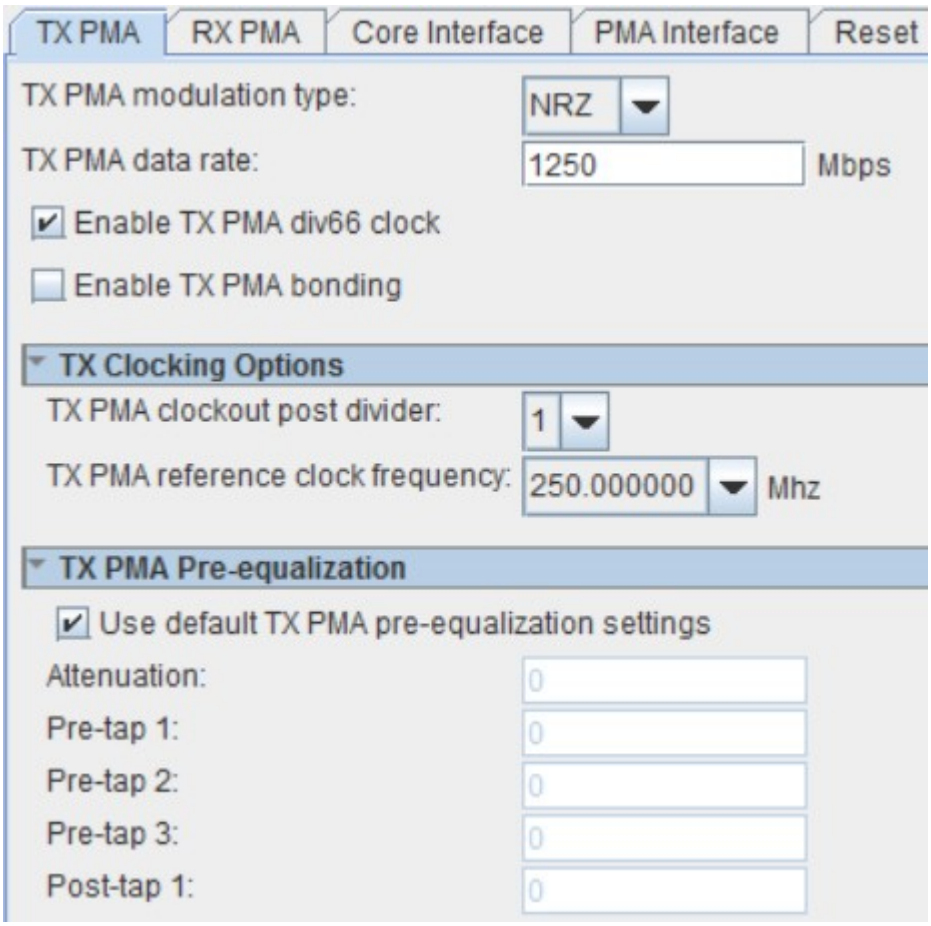


Table 13. TX PMA Options

Parameter	Value	Description
TX PMA modulation type	**NRZ** **PAM4**	Select the TX PMA modulation type based on your usage.
TX PMA data rate	For PAM4, **2000-57800 Mbps** For NRZ, **1000 - 28900 Mbps**	Specifies the transceiver TX data rate in units of megabits per second (Mbps). *Note:* To achieve 57.8 Gbps, you must select the **PMA direct high data rate PAM4** transceiver configuration rule.
Enable TX PMA div66 clock	**On/Off**	Enables or disables the TX PMA div 66 clock option.

continued...

Parameter	Value	Description
Enable TX PMA bonding	**On/Off**	Enables or disables the TX PMA bonding option. The parameter configures the reference clock, synchronizes the master connections, and so on. Deskew logic requires data to be marked with a toggle bit. Refer to *TX PMA Bonding*.
TX PMA clockout post divider	**1**, **2**, **4**, **8**	Specifies the post divider counter value for the `tx_pma_clkout` port.
TX PMA reference clock frequency	**125 - 700**	Selects the reference clock frequency options for the TX in MHz.

Related Information

TX PMA Bonding on page 94

2.2.2.2. TX PMA Pre-equalization

Table 14. TX PMA Pre-equalization

Parameter	Value	Description
Attenuation Value (VOD)	**0 to +26**	This is the range of attenuation (VOD).
Pre-tap 1	**-10 to +10**	This is the range of the first pre-tap pre-emphasis. Only even values are allowed.
Pre-tap 2	**-15 to +15**	This is the range of the second pre-tap pre-emphasis for both NRZ and PAM4 modes.
Pre-tap 3	**-1 to +1**	This is the range of the third pre-tap pre-emphasis for both NRZ and PAM4 modes.
Post-tap 1	**-18 to +18**	This is the range of the post-tap pre-emphasis. Only even values are allowed.
Use default TX PMA pre-equalization settings	**On/Off**	When enabled, the TX PMA pre-equalization settings are configured using Intel-specified default settings. When disabled, you must manually configure the TX PMA pre-equalization settings. Refer to the *TX Equalizer* section for more details.

Related Information

TX Equalizer on page 73

2.2.2.3. RX PMA Options

Figure 24. RX PMA Options

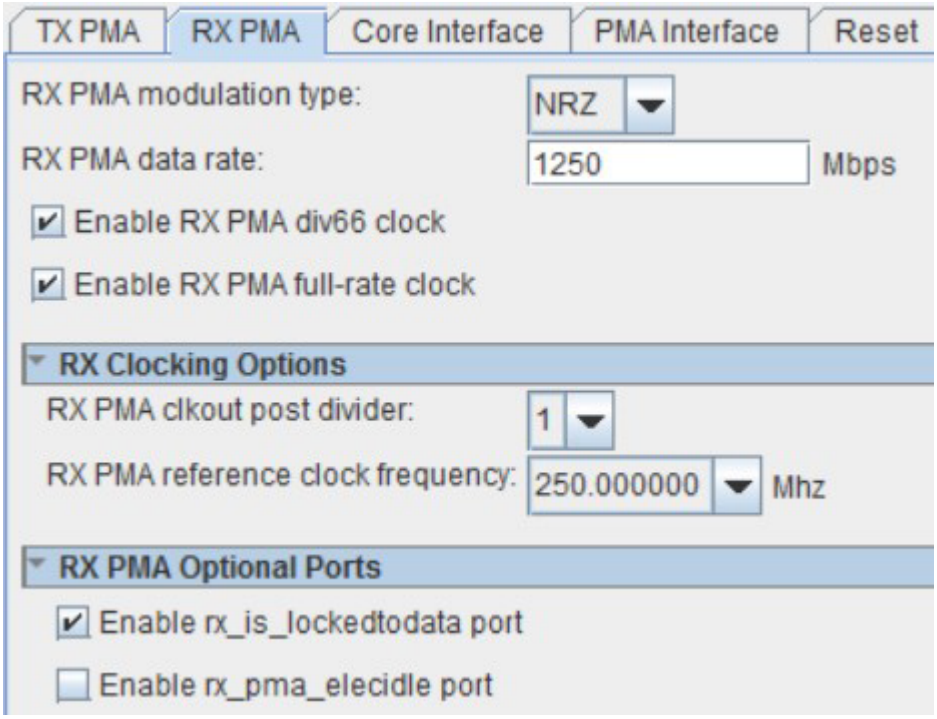
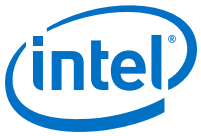


Table 15. RX PMA Options

Parameter	Value	Description
RX PMA modulation type	**NRZ, PAM4**	Select the RX PMA modulation type based on your usage.
RX PMA data rate	For PAM4, **2000 - 57800 Mbps** For NRZ, **1000 - 30000 Mbps**	Specifies the transceiver RX data rate in units of Mbps. *Note:* You must use the PAM4 high data rate mode for data rates > 30 Gbps.
Enable RX PMA div66 clock	**On/Off**	Enables or disables the RX PMA div 66 clock option.
Enable RX PMA full-rate clock	**On/Off**	Enables or disables the full-rate clock of the RX PMA.
RX PMA clockout post divider	**1, 2, 4, 8**	Specifies the post divider counter value for the `rx_pma_clkout` port.
RX PMA reference clock frequency	**100 - 700**	Selects the reference clock frequency options for the RX in MHz.



2.2.2.4. RX PMA Optional Ports

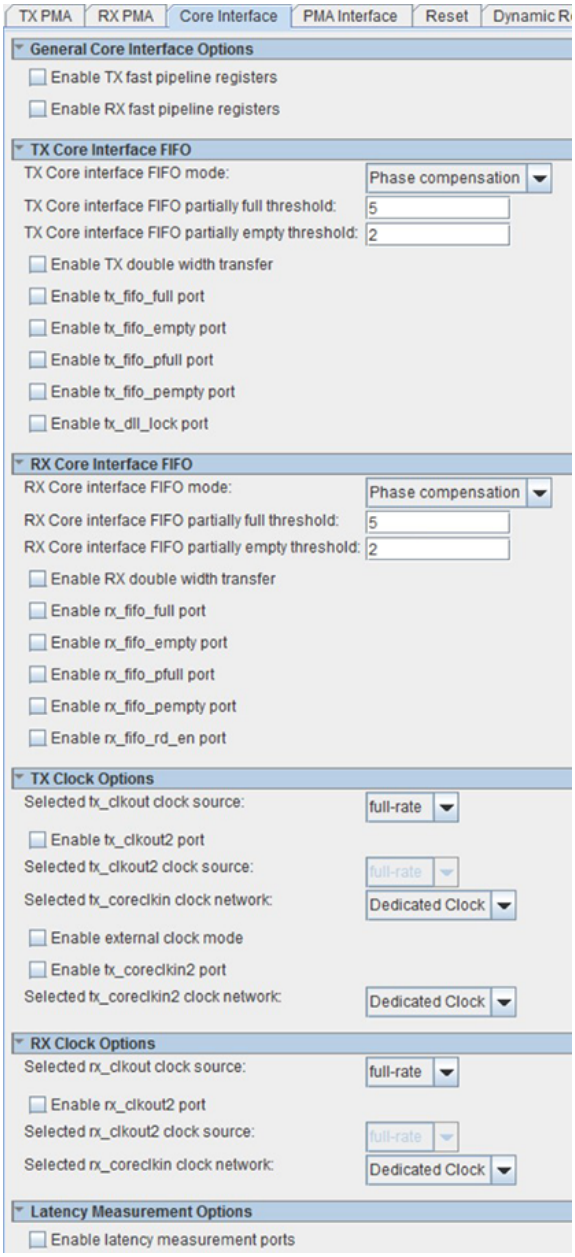
Table 16. RX PMA Optional Ports

Parameter	Value	Description
Enable rx_is_lockedtodata port	**On/Off**	Enables the optional `rx_is_lockedtodata` status output port. This signal indicates that the RX CDR is currently in lock to data mode, or is attempting to lock to the incoming data stream. This is an asynchronous output signal, and is also available as part of the E-Tile Native PHY register space.
Enable rx_pma_elecidle port	**On/Off**	

2.2.3. Core Interface Options

These Native PHY IP core parameters allow you to customize the transceiver-to-core interface.

Figure 25. Core Interface Options



2.2.3.1. Core Interface Parameters

The core interface is the interface between the transceiver EMIB and the FPGA core EMIB. You can use these options to customize the core interface.

Based on the transceiver configuration rule you select, the Native PHY IP core **Parameter Editor** reports error or warning messages if your settings violate the protocol standard.

Table 17. Core Interface Parameters

Parameter	Range	Description
General Core Interface Options		
Enable TX fast pipeline registers	**On/Off**	Enables the optional fast pipeline registers in the TX parallel datapath. The fast pipeline registers are hyper-registers, which are clocked by the `tx_coreclkin` input. You must clear fast pipeline registers synchronously. Enable this option to achieve better setup time for TX parallel data transfer from the FPGA core to the transceiver.
Enable RX fast pipeline registers	**On/Off**	Enables the optional fast pipeline registers in the RX parallel datapath. The fast pipeline registers are hyper-registers, which are clocked by the `rx_coreclkin` input. You must clear fast pipeline registers synchronously. Enable this option to achieve better setup time for RX parallel data transfer from the transceiver to the FPGA core.
TX Core Interface FIFO		
Enable TX double width transfer	**On/Off**	Enables or disables the TX transfer interface FIFO double width mode. Use this option when you need to divide the core frequency by two so as not to exceed the maximum EMIB frequency specifications. In duplex mode, select this parameter for both TX and RX simultaneously.
RX Core Interface FIFO		
Enable RX double width transfer	**On/Off**	Enables or disables the RX transfer interface FIFO double width mode. Use this option when you need to divide the core frequency by two so as not to exceed the maximum EMIB frequency specifications. In duplex mode, select this parameter for both TX and RX simultaneously.

2.2.3.2. TX Clock Options

Table 18. TX Clock Options

Parameter	Range	Description
Selected tx_clkout clock source	**Full-rate, half-rate, div66**	Specifies the clock source for the `tx_clkout` output clock.
Enable tx_clkout2 port	**On/Off**	Enables the optional `tx_clkout2` output clock.
Selected tx_clkout2 clock source	**Full-rate, half-rate, div66**	Specifies the clock source for `tx_clkout2` output clock after enabling the `tx_clkout2` port.
Selected tx_coreclkin clock network	**Dedicated Clock Global Clock**	Specifies the type of clock network to route the clock signal to the `tx_coreclkin` port. **Dedicated Clock** allows a higher maximum frequency (fmax) between the FPGA core and the transceiver. The number of dedicated clock lines are limited.
Enable tx_coreclkin2 port	**On/Off**	Enables the optional `tx_coreclkin2` input clock.
Selected tx_coreclkin2 clock network	**Dedicated Clock Global Clock**	Specifies the type of clock network to route the clock signal to the `tx_coreclkin2` port. **Dedicated Clock** allows a higher maximum frequency (fmax) between the FPGA core and the transceiver. The number of dedicated clock lines are limited.
Enable external clock mode	**On/Off**	Enables or disables the `tx_coreclkin2` input clock to drive the transfer clock.

2.2.3.3. RX Clock Options

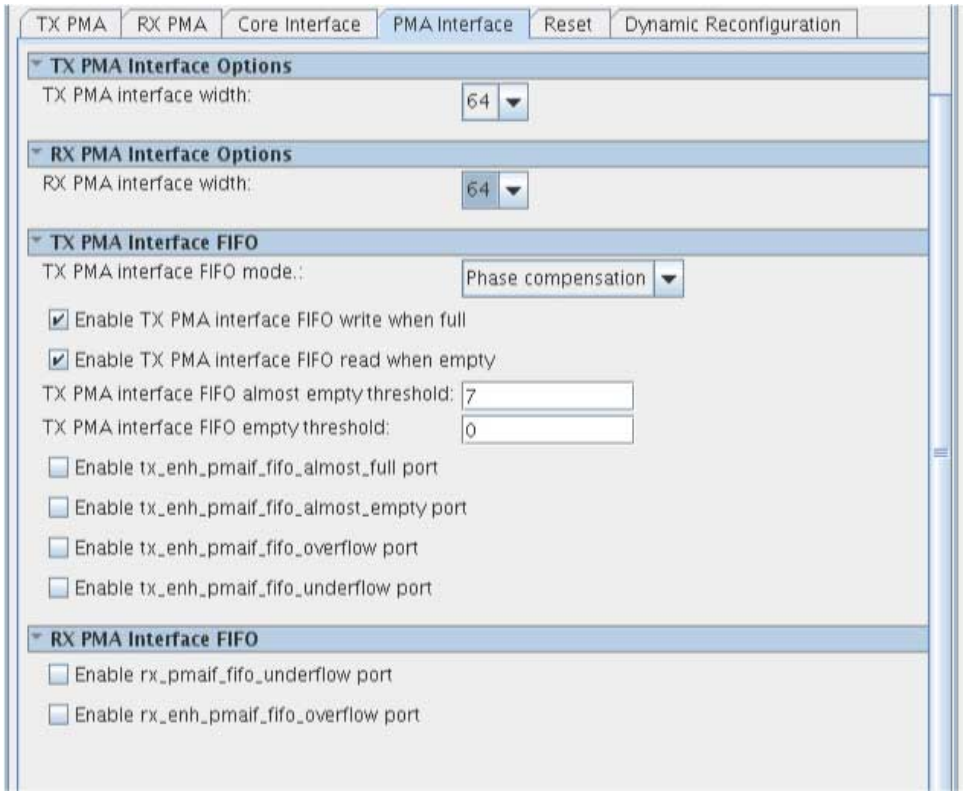
Table 19. RX Clock Options

Parameter	Range	Description
Selected rx_clkout clock source	**Full-rate, half-rate, div66**	Specifies the clock source for the `rx_clkout` output clock. • Full-rate: (data rate / PMA width) • Half-rate: (data rate / 2*PMA width) • Div66: (data rate / 66)
Enable rx_clkout2 port	**On/Off**	Enables the optional `rx_clkout2` output clock enabling the `rx_clkout2` port.
Selected rx_clkout2 clock source	**Full-rate, half-rate, div66**	Specifies the clock source for the `rx_clkout2` output clock. • Full-rate: (data rate / PMA width) • Half-rate: (data rate / 2*PMA width) • Div66: (data rate / 66)
Selected rx_coreclkin clock network	**Dedicated Clock** **Global Clock**	Specifies the type of clock network to route the clock signal to the `rx_coreclkin` port. **Dedicated Clock** allows a higher maximum frequency (fmax) between the FPGA core and the transceiver. The number of dedicated clock lines are limited. Intel recommends using dedicated clocks.

2.2.4. PMA Interface

PMA interface options are related to the interface of PMA side of the bridge between the PMA and the FPGA core, the FEC module, and so on.

Figure 26. PMA Interface Options



Related Information

Physical Medium Attachment (PMA) Architecture on page 70

2.2.4.1. PMA Interface Options

Table 20. PMA Interface Options

Parameter	Value	Description
TX PMA interface width	**16, 20, 32, 40, 64**	Specifies the TX data interface width of the PMA. Refer to *PMA Architecture* for mapping the data widths to PMA mode.
RX PMA interface width	**16, 20, 32, 40, 64**	Specifies the RX data interface width of the PMA. Refer to *PMA Architecture* for mapping the data widths to PMA mode.
TX PMA interface FIFO mode	**Phase Compensation** **Elastic**	Specifies the mode for the TX PMA Interface FIFO.
Enable TX PMA interface FIFO write when full	**On/Off**	Enables or disables the TX PMA interface FIFO write option when the FIFO is full.
Enable TX PMA interface FIFO read when empty	**On/Off**	Enables or disables the TX PMA interface FIFO read option when the FIFO is empty.
TX PMA interface FIFO almost empty threshold	**7**	Specifies the almost empty threshold for the TX PMA Interface FIFO.
TX PMA interface FIFO empty threshold	**0**	Specifies the empty threshold for the TX PMA Interface FIFO.
Enable tx_enh_pmaif_fifo_almost_full port	**On/Off**	Enables the port which indicates TX PMA Interface FIFO's almost full condition.
Enable tx_enh_pmaif_fifo_almost_empty port	**On/Off**	Enables the port which indicates TX PMA Interface FIFO's almost empty condition.
Enable tx_enh_pmaif_fifo_overflow port	**On/Off**	Enables the port which indicates TX PMA Interface FIFO's overflow condition. The port/signal is sticky, so you must reset it.
Enable tx_enh_pmaif_fifo_underflow port	**On/Off**	Enables the port which indicates TX PMA Interface FIFO's underflow condition. The port/signal is sticky, so you must reset it.
Enable rx_pmaif_fifo_underflow port	**On/Off**	Enables the port which indicates RX PMA Interface FIFO's underflow condition. The port/signal is sticky, so you must reset it.
Enable rx_enh_pmaif_fifo_overflow port	**On/Off**	Enables the port which indicates RX PMA Interface FIFO's overflow condition. The port/signal is sticky, so you must reset it.

Related Information

Physical Medium Attachment (PMA) Architecture on page 70

2.2.4.2. Gearbox 64/66

Because E-tile transceivers do not support 66 bit interface natively, you can use Gearbox 64/66 mode if the protocol design uses a 66 bit interface. You must use `data_valid` signals on both the TX and RX directions.

While configuring the transceiver Native PHY, you can specify parameters and modes from transceiver configuration rules given under **Datapath Options** in the **Parameter Editor**.

Table 21. Gearbox 64/66 Interface Ports

Parameter Name	Description
Enable tx_enh_pmaif_fifo_almost_full port	Indicates when the TX PMA interface FIFO is almost full. This signal is transferred over FSR.
Enable tx_enh_pmaif_fifo_almost_empty port	Indicates when the TX PMA interface FIFO is almost empty. This signal is transferred over SSR.
Enable tx_enh_pmaif_fifo_overflow port	Indicates when the TX PMA Interface FIFO has overflow. This signal is transferred over SSR.
Enable tx_enh_pmaif_fifo_underflow port	Indicates when the TX PMA interface FIFO has underflow. This signal is transferred over SSR.
Enable RX PMA interface FIFO almost full threshold	Indicates when the TX PMA interface FIFO is almost full. This signal is transferred over FSR.
Enable RX PMA interface FIFO almost empty threshold	Indicates when the TX PMA interface FIFO is almost empty. This signal is transferred over SSR.
Enable rx_pmaif_fifo_underflow port	Indicates when the RX PMA interface FIFO has underflow. This signal is transferred over FSR.
Enable rx_enh_pmaif_fifo_overflow port	Indicates when the RX PMA interface FIFO has overflow. This signal is transferred over SSR.
Enable TX bit reversal[4]	
Enable RX bit reversal[4]	
Enable TX sync header bit reversal[4]	
Enable RX sync header bit reversal[4]	
Enable tx_pmaif_bitslip port	Gearbox TX bitslip control
Enable rx_pmaif_bitslip port	Gearbox RX bitslip control
TX sync header location[4]	
RX sync header location[4]	

(4) For 66 bit parallel data to be transmitted, data[65:64] is considered sync header bits and data[63:0] is considered as data bits. Bit reversals are applied to sync header and data bits separately. The sync header location parameter takes effect after bit reversals. If 1-0 is selected for the sync header location, the sync header is placed into LSB, in other words, the data is rearranged as {data[63:0], data[65:64]}. If 65-64 is selected for the sync header location, the data is left untouched. Parameters on RX work similarly.

Restrictions:

- There are restrictions on power saving simplex modes.
- Configuring TX and RX data paths with different data rates is not supported.
- There is no user-defined read enable on the RX PMAIF FIFO.
- It is only supported in Interlaken with the use of an internal `.ini`. For more details, contact My Intel support.

Related Information

My Intel Support

2.2.5. PMA Adaptation Parameters

After power-on configuration, you may need to trigger PMA adaptation on the RX interface to achieve optimal performance. The PMA interface performs adaptation for initial adaptation and continuous adaptation which runs continuously in background.

Note: If you require more information about the PMA adaptation parameters, contact My Intel support. The definitions of the PMA adaptation parameters are not publicly available.

Figure 27. PMA Adaptation Options

You specify settings on the **PMA Adaptation** tab for both initial and continuous adaptation. PMA Adaptation is only available for E-tile variants.

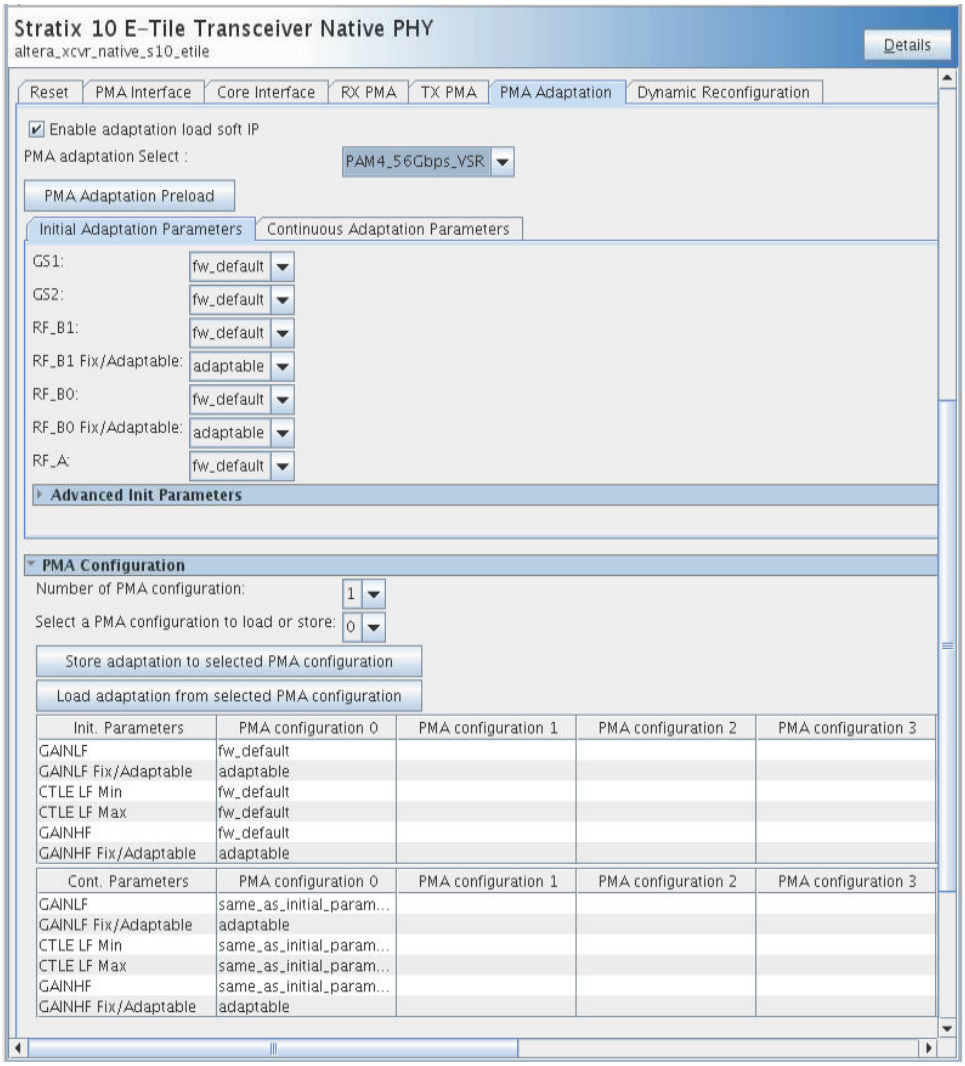


Table 22. PMA Adaptation

Parameter	Value	Description
Enable adaptation load soft IP	**On/Off**	Enables PMA adaptation parameter customization.
PMA adaptation select	**PAM4_56Gbps_LR** **PAM4_56Gbps_VSR** **NRZ_28Gbps_LR** **NRZ_28Gbps_VSR** **NRZ_10Gbps_25/15/10dB**	Selects preloaded PMA configuration parameters.

Table 23. Initial Adaptation Parameters

Parameter	Value	Description
GS1	**Fw_default, 0, 1, 2, 3**	CTLE Low Frequency Gain Shaping 1.
GS2	**Fw_default, 0, 1, 2, 3**	CTLE Low Frequency Gain Shaping 2.
RF_B1	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8**	RF_B1 Setting.
RF_B1 Fix/Adaptable	**Fixed, Adaptable**	RF_B1 Setting.
RF_B0	**Fw_default, 0, 1, 2, 3, 4, 5**	RF_B0 Setting.
RF_B0 Fix/Adaptable	**Fixed, Adaptable**	RF_B0 Setting.
RF_A	**Fw_default, 100, 110, 120, 130, 140, 150, 160**	RF_A Setting.
GAINLF	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	CTLE Low Frequency Gain.
GAINLF Fixed/ Adaptable	**Fixed, Adaptable**	CTLE Low Frequency Gain.
CTLE LF Min	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits CTLE LF minimum.
CTLE LF Max	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits CTLE LF maximum.
GAINHF	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Inputs CTLE High Frequency Gain.
GAINHF Fixed/ Adaptable	**Fixed, Adaptable**	Inputs CTLE High Frequency Gain - Fix or Adaptable options.
CTLE HF Min	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits CTLE HF minimum.
CTLE HF Max	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits CTLE HF maximum.
RF_P2	**Fw_default, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10**	RF_P2 setting.
RF_P2 Fixed/Adaptable	**Fixed, Adaptable**	RF_P2 setting - Fix or Adaptable options.
RF_P2 Min	**Fw_default, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10**	Limits RF_P2 minimum.
RF_P2 Max	**Fw_default, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10**	Limits RF_P2 maximum.
RF_P1	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	RF_P1 setting.
RF_P1 Fixed/Adaptable	**Fixed, Adaptable**	RF_P1 setting - Fix or Adaptable options.
RF_P1 Min	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits RF_P1 minimum.
RF_P1 Max	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Limits RF_P1 maximum.
RF_P1 Threshold	**Fw_default, -15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Controls the rate RF_P1 adapts.
RF_P0	**Fw_default, -15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Sets RF_P0.

continued...

Parameter	Value	Description
RF_P0 Fixed/Adaptable	**Fixed, Adaptable**	Limits RF_P0 - Fix or Adaptable options.
RF_P0 Threshold	**Fw_default, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15**	Controls the rate RF_P0 adapts.
RF_B0T	**Fw_default, 10, 15, 20, 25, 30, 35, 40, 45, 50**	Controls the rate RF_B0 adapts.

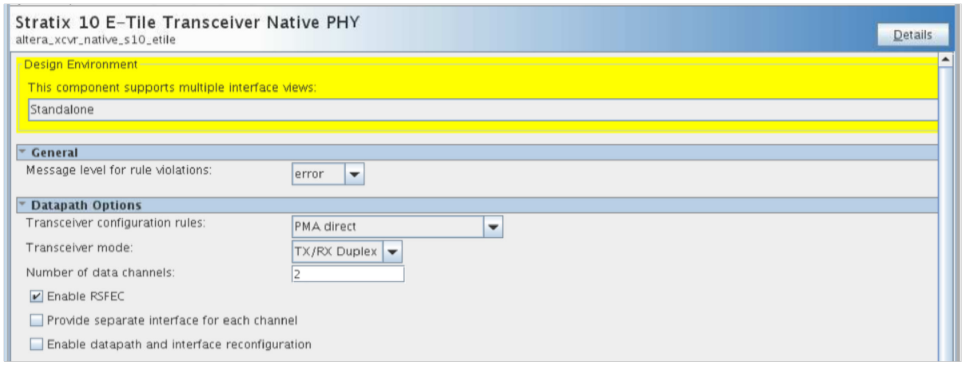
Related Information

My Intel Support

2.2.6. Reed Solomon Forward Error Correction (RS-FEC) Parameters

The Native PHY IP Core supports RS-FEC (528, 514) and (544, 514). You can enable this functionality by selecting the **Enable RS-FEC** option in the **Datapath Options** section of the GUI.

Figure 28. GUI with Enable RS-FEC Option Selected



RS-FEC merging is not supported with more than one Native PHY IP instance. A Native PHY IP with an RS-FEC block enabled uses the whole RS-FEC block, even if it uses only one channel in the IP. You can use dynamic reconfiguration to utilize the same RS-FEC block for implementation on different protocols.

The RS-FEC IP core supports the following modes:

- 32GFC/ CPRI 24G (x1, x2, x3, x4)
- 128 GFC
- 25GE FEC Direct (x1, x2, x3, x4)
- 100 GbE
- Interlaken [(5)]

(5) The Interlaken mode requires a special string in the quartus.ini file to enable its functionality. For more details, contact My Intel support.

Table 24. RS-FEC Mode Configurations

Supported Mode	RS-FEC Code Type	FEC Mode	Alignment/Scrambling/ Transcoder Mode	Transcoder Bypass
32GFC/ CPRI 24G (x1, x2, x3, x4)	NRZ (528, 514)	Fractured	Fibre channel	Not bypassed
128 GFC		Aggregate	Fibre channel	Not bypassed
25GE FEC Direct (x1, x2, x3, x4)		Fractured	Basic	Not bypassed
100 GbE		Aggregate	Basic	Not bypassed
Interlaken	PAM4 (544, 514)	Aggregate	Fibre channel	Bypassed

Figure 29. RS-FEC Options

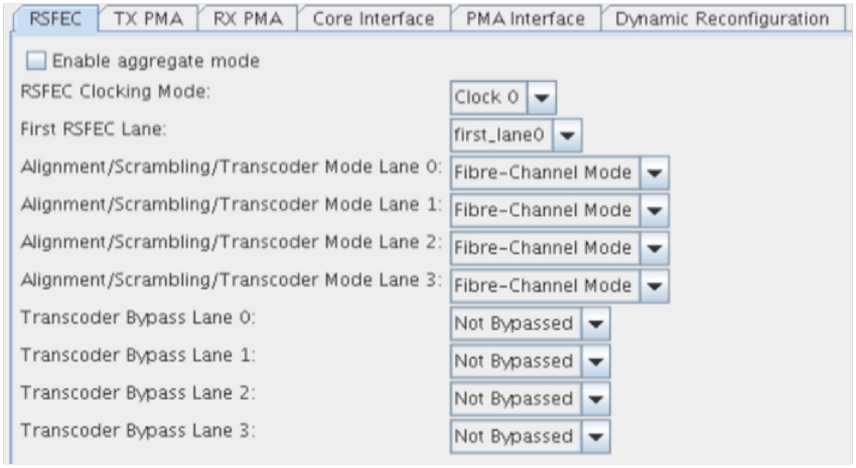


Table 25. RS-FEC Parameters

Parameter	Value	Description
Enable aggregate mode	**On/Off**	
RS-FEC Clocking Mode	**EHIP clock** **Clock 0** **Clock 1** **Clock 2** **Clock 3** **No Clock**	Sets the clocking mode for the RS-FEC block. For some RS-FEC topologies, the clock selection is fixed. In all other cases, this control selects the TX adapter clock used to clock the RS-FEC block.
First RS-FEC Lane	**first_lane0** **first_lane1** **first_lane2** **first_lane3**	Selects the first RS-FEC lane to be used. There are four lanes in the RS-FEC block. When the RS-FEC block is in fractured mode, any of the four lanes may be selected as the first lane. For multiple channel Native PHY IP core instances with the RS-FEC block enabled, the RS-FEC lanes used must be contiguous and must fit within a single four-channel RS-FEC block.
Alignment/Scrambling/Transcoder Mode	**Basic Mode** **Fibre-Channel Mode**	Basic Mode selects Ethernet-like mode for a specific lane. The transcoder is not bypassed, alignment or codeword markers are used for alignment of the RS-FEC block, and the PN-5280 scrambling is disabled.

continued...

Parameter	Value	Description
		Fibre-Channel Mode selects Fibre-Channel mode for RS-FEC lanes. Fibre-channel mode disables RS-FEC alignment/ codeword markers. Selecting Fibre-Channel mode also selects PN-5280 scrambling when in fractured mode. CPRI Mode is identical to Fibre-Channel mode with respect to the RS-FEC operation.
Transcoder Bypass Lane	**Not Bypassed** **Bypassed**	Selects the transcoder mode for RS-FEC lanes. • Not bypassed: The transcoder is not bypassed. • Bypassed: Bypasses the transcoder to enable a user-defined transcoder implemented in the FPGA.

Related Information

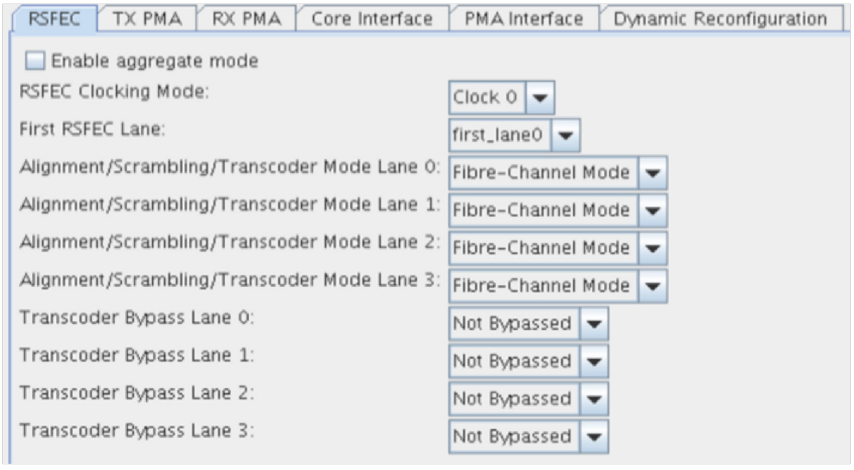
My Intel Support

2.2.6.1. Fibre-Channel and CPRI Modes

The core RS-FEC setup is identical for 32 GFC (Fibre-Channel) and CPRI. The RS-FEC options appear in the GUI after you select the **Enable RSFEC** option. In these modes alignment/codeword markers are not used, and PN-5280 scrambling and descrambling are used.

The **Number of data channels** indicates how many data channels are RS-FEC-enabled within one block. The **First RSFEC Lane** determines, which lane is the first lane within one block. Together, these two parameters determine and fix the lanes you are using in the IP core. For example, if **Number of data channels** is set to 3, the only valid option for the first lane is 0 or 1. All the lanes must be contiguously placed within one block, and you must observe this rule to achieve valid IP configurations. Once you have correctly set those two parameters, the GUI indicates the channels being enabled.

Figure 30. Fibre-Channel Mode Settings



2.2.6.2. 128 GFC Mode

In 128 GFC mode, fibre-channel is enabled for all lanes, and no additional settings are required. Scrambling or descrambling is disabled in 128 GFC mode. While the transcoder bypass setting options for the RS-FEC are available, they are not required for fibre-channel mode. The RS-FEC must be in aggregate mode for 128 GFC.

When you select the aggregate mode for 128 GFC, you must select the **TX/RX reset sequencing** option on the **Reset** tab. The GUI indicates an error unless you enable this option.

When the RS-FEC aggregate mode is enabled, the Native PHY IP core supports the 128 GFC mode. The other RS-FEC settings are the same for the aggregated 128 GFC mode and the fractured 32 GFC mode. The signaling data rate for 32 GFC is 28.05 Gbps. For 128 GFC, the aggregate signaling data rate is 112.2 Gbps (four lanes of 28.05 Gbps).

Figure 31. 128 GFC Mode RS-FEC Settings

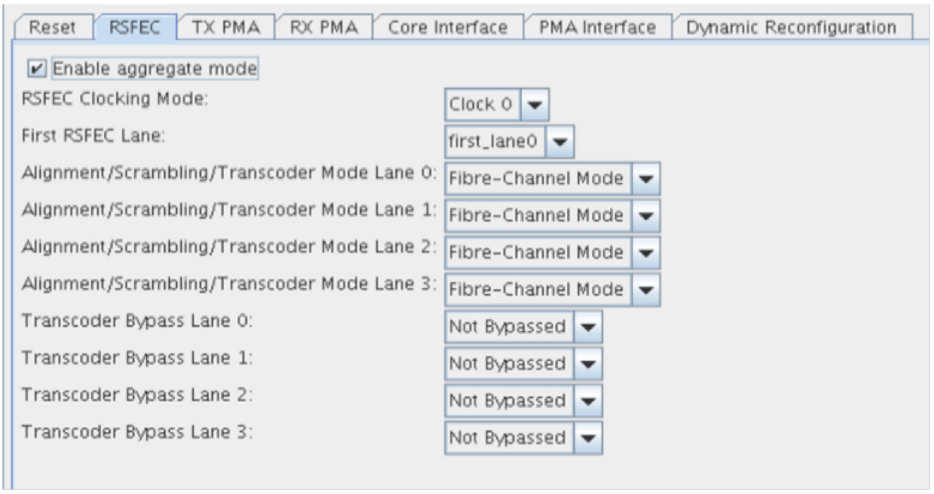
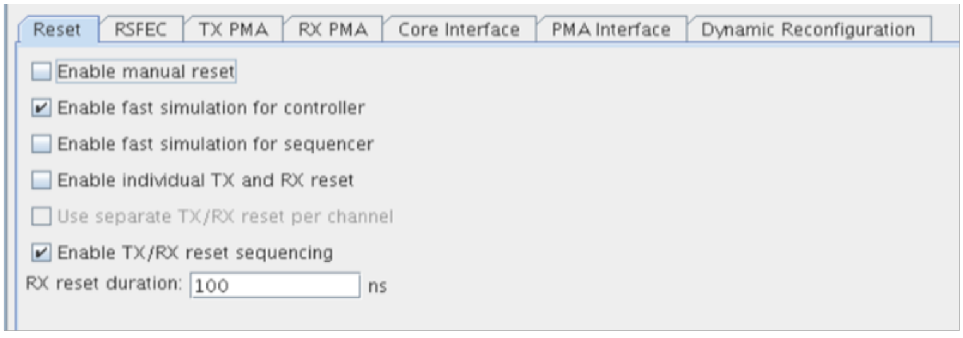


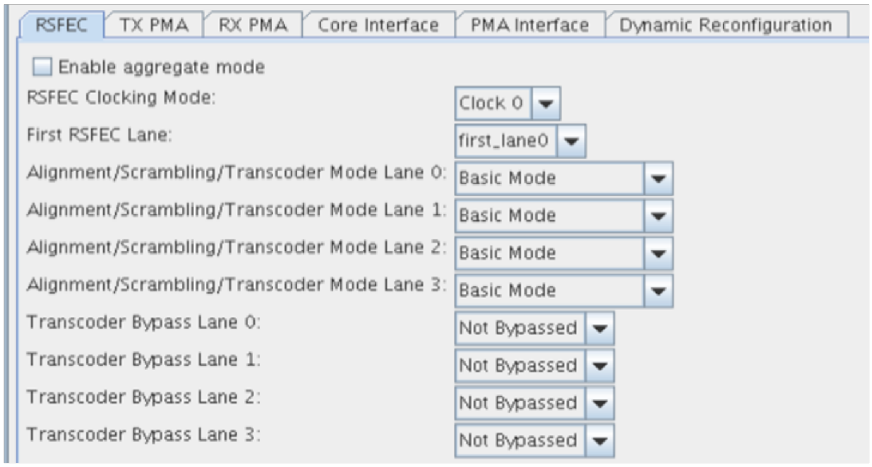
Figure 32. 128 GFC Mode Reset Settings



2.2.6.3. 25 GbE FEC Direct Mode

Figure 33. 25 GbE FEC Direct Mode Settings

The RS-FEC is enabled in fractured mode.



2.2.6.4. Interlaken Mode

The RS-FEC is available in 100G Interlaken mode. In the 100G Interlaken mode, you must set the RS-FEC to aggregate mode across all four lanes with transcoder bypass enabled. The modulation scheme for this mode is PAM4.

Note: You must enable a special **Quartus.ini** file to use this mode. For more details, contact My Intel support.

Related Information

My Intel Support

2.2.7. Reset Parameters

The Native PHY IP Core reset parameters provide reset control for the PMA interface and adapter.

Figure 34. Reset Options

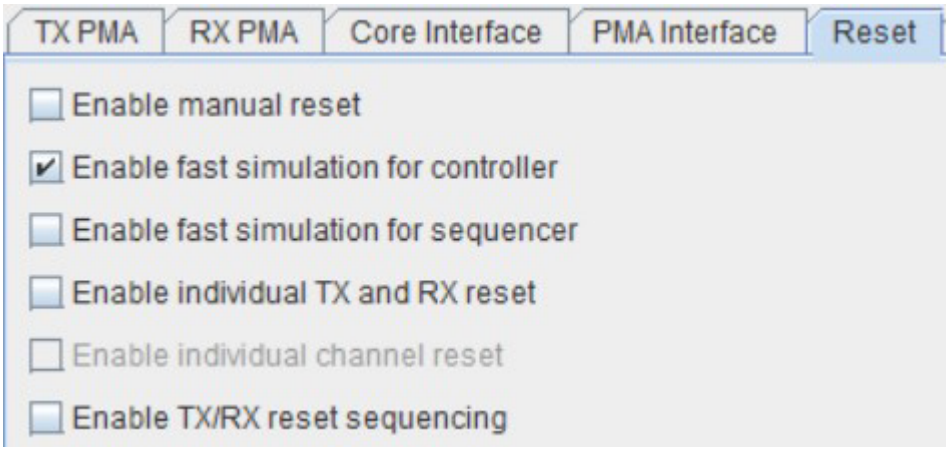


Table 26. Reset Parameters

Parameter	Value	Description
Enable manual reset	**On/Off**	When enabled, sets manual reset mode. You must control all reset signals for the device.
Enable fast simulation for controller	**On/Off**	When enabled, the IP uses reduced reset time for reset controller in simulation.
Enable fast simulation for sequencer	**On/Off**	When enabled, the IP disables reset staggering in simulation. The reset behavior in simulation is different from the reset behavior in hardware.
Enable individual TX and RX reset	**On/Off**	When enabled, the IP uses `tx_reset` and `rx_reset` input ports to control TX and RX individually, otherwise uses `reset` input ports to control both TX and RX.
Enable individual channel reset	**On/Off**	When enabled, you can control the channels individually.
Enable TX/RX reset sequencing	**On/Off**	When enabled, the IP staggers the deassertion of the TX reset before the RX reset. That is, `tx_reset` deassertion gates `rx_reset` deassertion.

2.2.8. Dynamic Reconfiguration Parameters

Dynamic reconfiguration is the process of modifying transceiver channels to meet changing requirements during device operation.

You can customize channels by triggering reconfiguration during device operation or after power-up.

Figure 35. Dynamic Reconfiguration Options

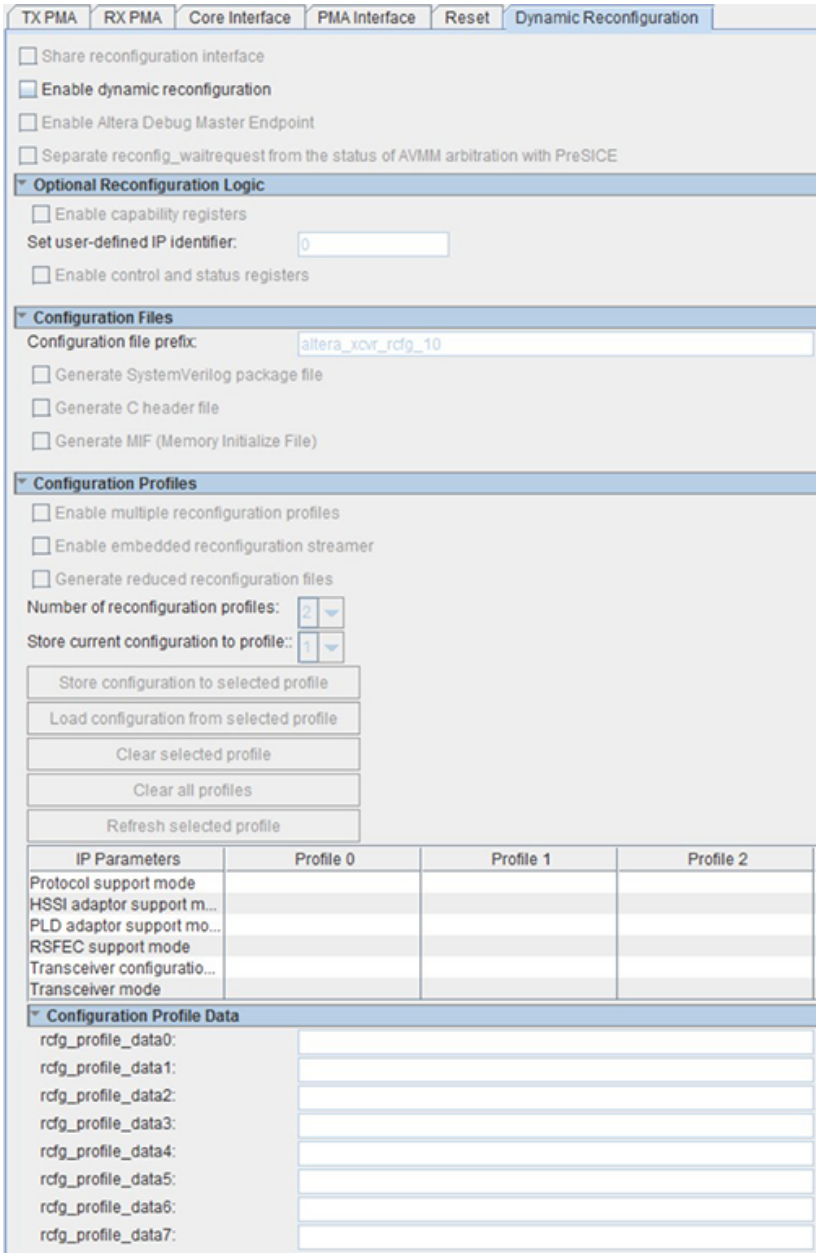


Table 27. Dynamic Reconfiguration Parameters

Parameter	Value	Description
Enable dynamic reconfiguration	**On/Off**	Enables the dynamic reconfiguration interface.
Share reconfiguration interface	**On/Off**	When enabled, the Native PHY presents a single Avalon-MM (AVMM) slave interface for dynamic reconfiguration of all channels. In this configuration, the upper [n-1:19] address bits of the reconfiguration address bus specify the selected channel,

continued...

Parameter	Value	Description
		where 'n' is the log base 2 of the number of channels. Address bits [18:0] provide the register offset address within the reconfiguration space of the selected channel.
Enable Native PHY Debug Master Endpoint	**On/Off**	When enabled, the Native PHY includes an embedded Native PHY Debug Master Endpoint (NPDME) that connects internally to the AVMM slave interface. The NPDME accesses the reconfiguration space of the transceiver and performs certain test and debug functions via JTAG using the **System Console**. This option requires you to enable the **Share reconfiguration interface** option for configurations using more than one channel and may also require that you include a `jtag_debug` link in the system.
Separate reconfig_waitrequest from the status of AVMM arbitration	**On/Off**	When enabled, the `reconfig_waitrequest` does not indicate the status of AVMM arbitration. The AVMM arbitration status is reflected in a soft status register bit. This feature requires that you enable the **Enable control and status registers** feature under **Optional Reconfiguration Logic**. The default setting is **Off**.
Enable capability registers	**On/Off**	Enables capability registers, which provide high level information about the transceiver configuration. Refer to the *PMA Capability Registers* section.
Set user-defined IP identifier	**0**	Sets a user-defined numeric identifier that can be read from the `user_identifer` offset when the capability registers are enabled.
Enable control and status registers	**On/Off**	Enables soft registers for reading status signals and writing control signals on the PHY interface through the embedded debug. Refer to the *PMA/PMA Interface AVMM Registers* section.
Configuration file prefix	**File prefix**	Specifies the file prefix to use for generated configuration files when enabled. Each variant of the IP should use a unique prefix for configuration files.
Generate SystemVerilog package file	**On/Off**	When enabled, The IP generates a SystemVerilog package file named `[Configuration file prefix]_reconfig_parameters.sv` containing parameters defined with the attribute values needed for reconfiguration.
Generate C header file	**On/Off**	When enabled, The IP generates a C header file named `[Configuration file prefix]_reconfig_parameters.h` containing macros defined with the attribute values needed for reconfiguration.
Generate MIF (Memory Initialize File)	**On/Off**	When enabled, the IP generates a Memory Initialization File (MIF) named `[Configuration file prefix]_reconfig_parameters.mif`. The MIF file contains the attribute values needed for reconfiguration in a data format.
Enable multiple reconfiguration profiles	**On/Off**	When enabled, you can use the GUI to store multiple configurations. The IP generates reconfiguration files for all of the stored profiles. The IP also checks your multiple reconfiguration profiles for consistency to ensure you can reconfigure between them.
Enable embedded reconfiguration streamer	**On/Off**	Enables the embedded reconfiguration streamer, which automates the dynamic reconfiguration process between multiple predefined configuration profiles.
Generate reduced reconfiguration files	**On/Off**	When enabled, the Native PHY generates reconfiguration report files containing only the attributes or RAM data that are different between the multiple configured profiles.

continued...

Parameter	Value	Description
Number of reconfiguration profiles	**1 - 8**	Specifies the number of reconfiguration profiles to support when multiple reconfiguration profiles are enabled.
Store current configuration to profile	**On/Off**	Selects which reconfiguration profile to store when clicking the **Store profile** button.
rcfg_profile_data0	**Profile data**	Dynamic reconfiguration parameter data for Profile 0.
rcfg_profile_data1	**Profile data**	Dynamic reconfiguration parameter data for Profile 1.
rcfg_profile_data2	**Profile data**	Dynamic reconfiguration parameter data for Profile 2.
rcfg_profile_data3	**Profile data**	Dynamic reconfiguration parameter data for Profile 3.
rcfg_profile_data4	**Profile data**	Dynamic reconfiguration parameter data for Profile 4.
rcfg_profile_data5	**Profile data**	Dynamic reconfiguration parameter data for Profile 5.
rcfg_profile_data6	**Profile data**	Dynamic reconfiguration parameter data for Profile 6.
rcfg_profile_data7	**Profile data**	Dynamic reconfiguration parameter data for Profile 7.

Related Information

- PMA Capability Registers on page 211
- PMA AVMM Registers on page 213

2.2.9. Deskew Logic

TX Deskew Logic

Once bonding is enabled (or, in PMA direct high data rate mode, even if bonding is not enabled) deskew logic in the transceiver interface is engaged, which aligns the data that is transferred across multiple channels within the same clock cycle. However, deskew logic requires action on your part. Bit 33 of the TX parallel data, data[33], is mapped to function as the deskew pulse. For PMA Direct high data rate, you must drive the deskew pulse bit of all bonded channels with a pulse that is active on every eighth parallel clock cycle. The deskew logic uses the deskew pulse to align the FIFO. It takes several cycles for the channels to be aligned. You must perform an AVMM read to the TX deskew status register `cfg_tx_deskew_sts` of all bonded lanes to find out if deskew completed successfully which indicates that all bonded channels have aligned parallel data. The deskew status register also provides further information for debugging if deskew is not successful.

`cfg_tx_deskew_sts[2]`

- 0 = not aligned or not enabled or did not receive a deskew bit
- 1 = aligned

`cfg_tx_deskew_sts[1:0]`

- 00 = not yet received a deskew bit
- 01 = not aligned
- 10 = received one set of aligned deskew bits
- 11 = received 16 sets of aligned deskew bits

The deskew mechanism runs continuously. In other words, if the alignment lock is lost, monitoring `cfg_tx_deskew_sts` informs you about the status. The deskew mechanism works the same way for **PMA direct high data rate PAM4** mode for two EMIB channels. In other words, you must send deskew pulses for the data you sent to two EMIBs and at the master PMA interface you are aligned to before sending to a single PMA. In double width mode, the deskew pulse needs to be sent every fourth half clock cycle.

Figure 36. Deskew Pulse with Double Width Mode Off (Full-Rate)

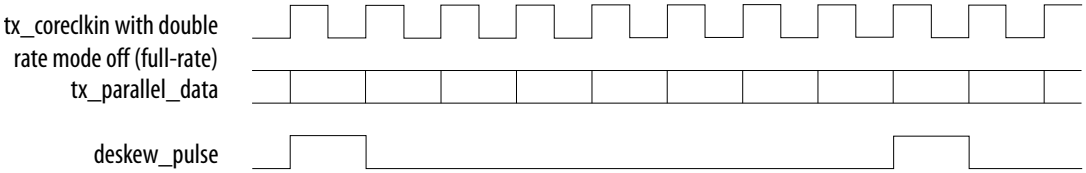
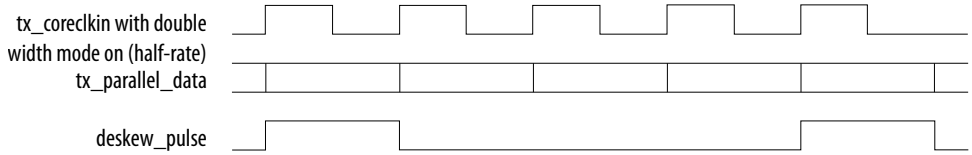


Figure 37. Deskew Pulse with Double Width Mode On (Half-Rate)



For the extra data bit's detailed usage, contact My Intel support.

E-Tile Native PHY IP Mode	TX/RX PMA Interface Width	Enable TX/RX Double Width Transfer	Valid Parallel Data	Deskew Bits	Pulse Clock Cycle
PMA Direct	16	No	Data [15:0]	Data[33]	8th cycle
PMA Direct	20	No	Data[19:0]	Data[33]	8th cycle
PMA Direct	32	No	Data[31:0]	Data[33]	8th cycle
PMA Direct	40	No	Data[39:0]	Not supported	Not supported
PMA Direct	16	Yes	Data[55:40] Data[15:0]	Data[33]	4th cycle
PMA Direct	20	Yes	Data[59:40] Data[19:0]	Data[33]	4th cycle
PMA Direct	32	Yes	Data[71:40] Data [31:0]	Data[33]	4th cycle
PMA direct high data rate PAM4	64	No	Data[111:80] Data[31:0]	Data[33] Data[113]	8th cycle
PMA direct high data rate PAM4	64	Yes	Data[151:120] Data[71:40] Data[111:80] Data[31:0]	Data[33] Data[113]	4th cycle

RX Deskew Logic

For PAM4 dual channel mode, the data comes from two EMIBs, so there can be skews in between. To mitigate this, the Native PHY IP implements a deskew function on RX side to align the two EMIBs' data. However, you can only enable this deskew function when the Native PHY IP is configured in **PMA direct high data rate PAM4** mode. The

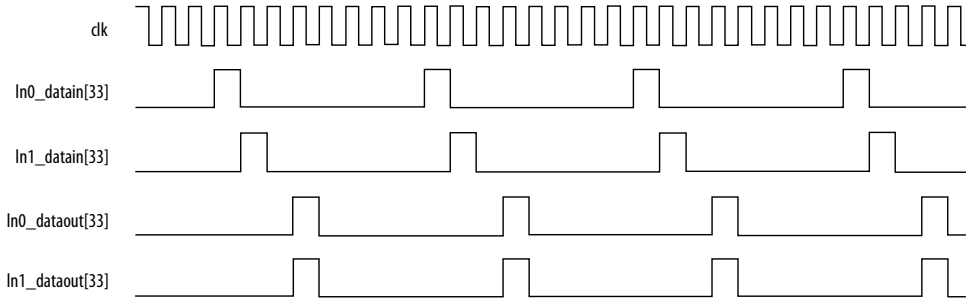
deskew logic samples the deskew bit of `rx_parallel_data` to detect if there is any misalignment. If there is misalignment, the deskew logic calculates skew cycles and outputs the aligned data.

The deskew logic can mitigate a maximum of two skew cycles for single width transfer and a maximum of one skew cycle for double width transfer. When the deskew logic is enabled, there is an added latency of up to three clock cycles even if there is no skew and an added latency of up to five clock cycles if there is two skew cycles.

To observe the misalignment, read the deskew bit in `rx_parallel_data[159:0]`, which is divided into lane 0 (`ln0` or `rx_parallel_data [79:0]`) and lane 1 (`ln1` or `rx_parallel_data[159:80]`).

If it is single data width transfer, the deskew bit is bit [33].

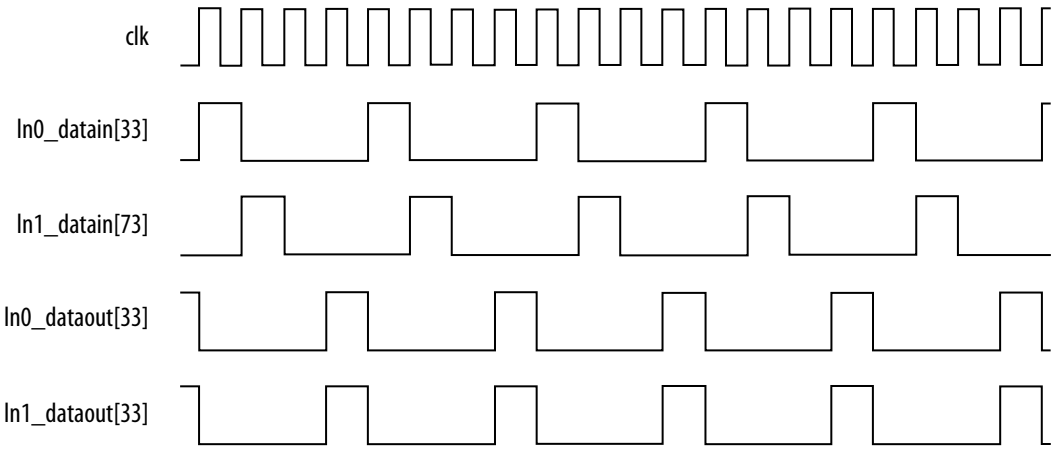
Figure 38. Single Data Width Transfer



If it is double data width transfer, both deskew bits [33] and [73] are valid. However, one appears in each channel, so there are four possible combinations:

- `ln0[33]` and `ln1[33]`
- `ln0[33]` and `ln1[73]`
- `ln0[73]` and `ln1[33]`
- `ln0[73]` and `ln1[73]`

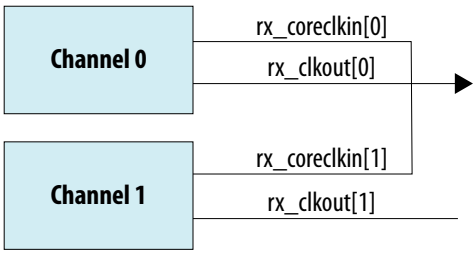
Figure 39. Double Data Width Transfer



RX Deskew Feature Requirements

- The Native PHY IP must be configured in **PMA direct high data rate PAM4** mode.
- The two EMIB channel clocks must be tied together to run on one clock domain.

Figure 40. Two EMIB Channel Clocks Working in the Same Clock Domain

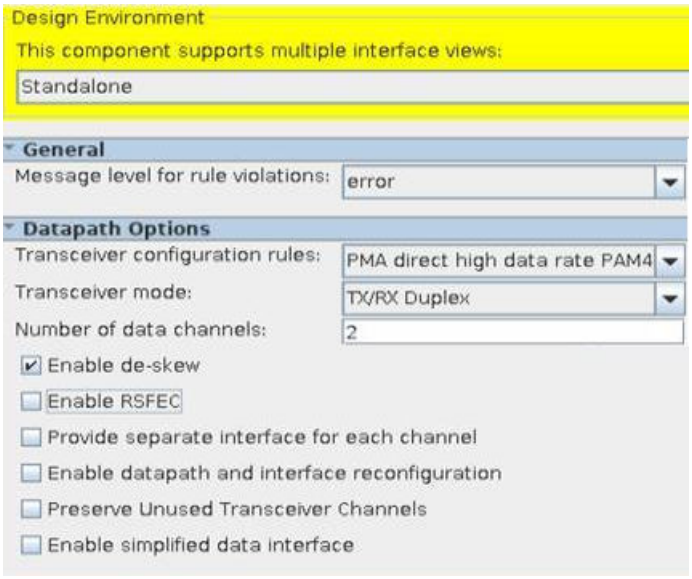


- `rx_ready` must show that the receiver is ready. If the receiver is not ready, the data is neither reliable nor ready. So you must reset the reset controller until `rx_ready` indicates that the receiver is ready.

Enabling RX Deskew Logic

1. From Native PHY IP GUI, select from **Transceiver configuration rules ➤ PMA direct high data rate PAM4**.
2. Turn on **Enable de-skew**.
3. Because RX deskew logic is applicable for **PMA direct high data rate PAM4** only, turn off **Enable RSFEC**.
4. Verify that your data is aligned by confirming that `rx_dskw_ready` is asserted.

Figure 41. Enabling RX Deskew Logic



Related Information

My Intel Support

2.2.10. Port Information

Table 28. Port Information

Port Name	Direction	Width	Description
pll_refclk0	Input	1 bit for each channel	Reference clock for the transceiver.
reset	Input	1 bit for each channel	Reset signal for the transceiver.
rx_serial_data	Input	1 bit for each channel	Positive signal for the receiver.
rx_serial_data_n	Input	1 bit for each channel	Negative signal for the receiver.
tx_serial_data	Output	1 bit for each channel	Positive signal for the transmitter.
tx_serial_data_n	Output	1 bit for each channel	Negative signal for the transmitter.
rx_parallel_data	Output	80 bits for each channel	Parallel data of the receiver side. Refer to Table 29 on page 60.
tx_parallel_data	Input	80 bits for each channel	Parallel data of the transmitter side. Refer to Table 29 on page 60.
tx_pma_ready	Output	1 bit for each channel	Ready status signal of the transmitter PMA.
tx_ready	Output	1 bit for each channel	Ready status signal of the transmitter.
rx_dskw_ready	Output	1 bit for each PMA channel	Ready status signal indicating that the deskew calculations are done and the data is available.
rx_pma_ready	Output	1 bit for each channel	Ready status signal of the receiver PMA.
rx_ready	Output	1 bit for each channel	Ready status signal of the receiver.
rx_is_lockedtodata	Output	1 bit for each channel	Locked to data status signal of the receiver.
rx_pma_elecidle	Input	1 bit for each channel	Electrical idle status signal of the receiver PMA.
rx_fifo_empty	Output	1 bit for each channel	Status signal indicating the receiver core interface FIFO is empty.
rx_fifo_full	Output	1 bit for each channel	Status signal indicating the receiver core interface FIFO is full.
rx_fifo_pempty	Output	1 bit for each channel	Status signal indicating the receiver core interface FIFO is partially empty.
rx_fifo_pfull	Output	1 bit for each channel	Status signal indicating the receiver core interface FIFO is partially full.
rx_fifo_rd_en	Input	1 bit for each channel	This port is used for Elastic FIFO mode. Asserting this signal enables the read from RX core FIFO.
tx_dll_lock	Output	1 bit for each channel	TX DLL locked status signal for data transfer.

continued...

Port Name	Direction	Width	Description
tx_fifo_empty	Output	1 bit for each channel	Status signal indicating the transmitter core interface FIFO is empty.
tx_fifo_full	Output	1 bit for each channel	Status signal indicating the transmitter core interface FIFO is full.
tx_fifo_pempty	Output	1 bit for each channel	Status signal indicating the transmitter core interface FIFO is partially empty.
tx_fifo_pfull	Output	1 bit for each channel	Status signal indicating the transmitter core interface FIFO is partially full.
latency_sclk	Input	1 bit for each channel	Clock signal for latency measurement(6) of the deterministic latency application.
rx_dl_async_pulse	Output	1 bit for each channel	Asynchronous output pulse signal for the receiver latency measurement(6) of the deterministic latency application. There is a start pulse and a stop pulse.
rx_dl_measure_sel	Input	1 bit for each channel	Mux select signal for the receiver latency measurement(6). 1 is for the datapath latency. 0 is for the wire delay.
tx_dl_async_pulse	Output	1 bit for each channel	Asynchronous output pulse signal for the transmitter latency measurement(6) of the deterministic latency application. There is a start pulse and a stop pulse.
tx_dl_measure_sel	Input	1 bit for each channel	Mux select signal for the transmitter latency measurement(6). 1 is for the datapath latency. 0 is for the wire delay.
tx_clkout	Output	1 bit for each channel	Clock output from the transmitter. You can select the full-rate, half-rate, or div66 option in the Native PHY GUI.
tx_clkout2	Output	1 bit for each channel	2nd clock output from the transmitter. You can select the full-rate, half-rate, or div66 option in the Native PHY GUI when the port is enabled.
tx_coreclkin	Input	1 bit for each channel	Transfer clock between the FPGA core and the transmitter.
tx_coreclkin2	Input	1 bit for each channel	2nd transfer clock between the FPGA core and the transmitter.
rx_clkout	Output	1 bit for each channel	Clock output from the receiver. You can select the full-rate, half-rate, or div66 option in the Native PHY GUI.
rx_clkout2	Output	1 bit for each channel	2nd clock output from the receiver. You can select the full-rate, half-rate, or div66 option in the Native PHY GUI when the port is enabled.
rx_coreclkin	Input	1 bit for each channel	Transfer clock between the FPGA core and the receiver.
rsfec_avmm2_avmmread_in	Input	1 bit	AVMM read signal of the AVMM2 interface for FEC.

continued...

(6) See the "Latency Measurement" section in the *E-tile Hard IP User Guide: E-tile Hard IP for Ethernet and E-Tile CPRI PHY Intel FPGA IPs*.

Port Name	Direction	Width	Description
rsfec_avmm2_avmmrequest_in	Input	1 bit	AVMM request signal of the AVMM2 interface for FEC.
rsfec_avmm2_avmmwrite_in	Input	1 bit	AVMM write signal of the AVMM2 interface for FEC.
rsfec_signal_ok[0]	Input	1 bit for each channel	Indicator to RS-FEC per lane that the PMA lane is up and stable(7)
rsfec_signal_ok[1]	Input	1 bit for each channel	
rsfec_signal_ok[2]	Input	1 bit for each channel	
rsfec_signal_ok[3]	Input	1 bit for each channel	
i_rsfec_pld_ready	Input	1 bit	Indicator to RS-FEC that the FPGA core and application layer is ready to start sending and receiving traffic.
reconfig_clk	Input	1 bit	Clock signal of reconfiguration interface.
reconfig_reset	Input	1 bit	Reset signal of reconfiguration interface.
reconfig_write	Input	1 bit	Write signal of reconfiguration interface.
reconfig_read	Input	1 bit	Read signal of reconfiguration interface.
reconfig_address	Input	19 bit	Address signal of reconfiguration interface (the upper [n-1:19] address bits of the reconfiguration address bus specify the selected channel, where 'n' is the log base 2 of the number of channels).
reconfig_writedata	Input	8 bit	Write data of reconfiguration interface.
reconfig_readdata	Output	8 bit	Read data of reconfiguration interface.
reconfig_waitrequest	Output	1 bit	Wait Request signal of reconfiguration interface.

Table 29. Parallel Data

E-Tile Native PHY Mode	TX/RX PMA Interface Width	Enable TX/RX double width transfer	Valid Parallel Data	Note
PMA Direct	16	No	Data [15:0]	N/A
PMA Direct	20	No	Data [19:0]	N/A
PMA Direct	32	No	Data [31:0]	N/A
PMA Direct	40	No	Data [39:0]	N/A
PMA Direct	16	Yes	Data [55:40] Data [15:0]	Data [55:40] is the first data group. Data [15:0] is the second data group.

continued...

(7) In **PMA direct high data rate PAM4** mode, when 2x is selected for the number of channels, on the transceiver side, only even indexed channels are actively sending and receiving data; odd indexed channels are powered down and not usable for other purposes. Nevertheless, you must assert rsfec_signal_ok for all 2x even and odd indexed channels.

E-Tile Native PHY Mode	TX/RX PMA Interface Width	Enable TX/RX double width transfer	Valid Parallel Data	Note
PMA Direct	20	Yes	Data [59:40] Data [19:0]	Data [59:40] is the first data group. Data [19:0] is the second data group.
PMA Direct	32	Yes	Data [71:40] Data [31:0]	Data [71:40] is the first data group. Data [31:0] is the second data group.
PMA Direct high data rate PAM4	64	No	Data [111:80] Data [31:0]	Data [31:0] is the lower bits data. Data [111:80] is the upper bits data.
PMA Direct high data rate PAM4	64	Yes	Data [151:120] Data [71:40] Data [111:80] Data [31:0]	Data [111:80] and Data [31:0] are the first data group. In this group, Data [31:0] is the lower bits data. Data [111:80] is the upper bits data. Data [151:120] and Data [71:40] are the second data group. In this group, Data [71:40] is the lower bits data. Data [151:120] is the upper bits data.

Bit Mapping for Native PHY TX and RX Datapaths

When the RS-FEC is enabled in the Native PHY IP, it is instantiated in the Native PHY IP netlist, and certain interface restrictions are required. In particular, the EMIB adapter FIFOs must be set to double-width mode, which means the TX and RX parallel datapaths are both 80 bits wide.

Also, it is necessary to clock the input side of the TX FIFO (80 bits wide) and the output side of the RX FIFO (80 bits wide) with a half-rate clock. Alternatively, the FIFO full and FIFO empty flags can be used to pace the FIFO FPGA interface.

Because of the double-width Native PHY IP datapath interface, a mapping is required from the datapath ports accessible in the Native PHY IP core to the datapath. The mapping is shown in the table below.

Table 30. 80 Bit Data Native PHY IP Double-Width TX/RX Ports

Bits	tx_parallel_data	rx_parallel_data
79	word_marking_bit_msb	word_marking_bit_msb
78	DESKEW	DESKEW
77	SNAPSHOT	RX_FIFO_USED[4:0]
76		
75		
74		
73		
72		RX_FIFO_EMPTY
71		RX_FIFO_PFULL
70		
69	SYNC	SYNC
68	VALID	VALID

continued...

67		
[66:40]	TXDATA[65:39]	RXDATA[65:39]
39	word_marking_bit_lsb	word_marking_bit_lsb
[38:0]	TXDATA[38:0]	RXDATA[38:0]

The word marking bits are inserted automatically by the IP, so you do not need to do anything with these bits. You just need to map your data to the Native PHY IP TX and RX ports as shown.

Legend for the above table:

- DESKEW: Deskew marker for each lane
- RX_FIFO_EMPTY: RX FIFO empty status from the PMA interface
- RX_FIFO_PFULL: RX FIFO partially full status from the PMA interface
- SYNC (TX and RX): Data to PCS synchronization (alignment/codeword marker or 257b synchronization)
- VALID:
  - TX: Deassert the valid line once every 33 cycles
  - RX: Data received from RS-FEC valid
- RX_FIFO_USED[4:0]:
  - [0]: PMA interface TX FIFO almost empty
  - [1]: PMA interface TX FIFO partially full
  - [2]: PMA interface TX FIFO underflow
  - [3]: PMA interface TX FIFO overflow
  - [4]: PMA interface RX FIFO overflow
- SNAPSHOT: Snapshot of the register counters. The rising edge of this signal latches running 64-bit counters into 32-bit registers. When 0, the registers are constantly being updated.

Related Information

"Latency Measurement" section of the *E-tile Hard IP User Guide: E-tile Hard IP for Ethernet and E-Tile CPRI PHY Intel FPGA IPs*

2.2.11. PLL Mode

PLL mode is a configuration of the E-tile transceiver Native PHY IP core that configures the E-tile transceiver as a PLL. It is used for external EMIB clocking configurations (see the use case in *Four 25 Gbps PMA Direct Channel (with FEC) within a Single FEC Block*). It does not support dynamic reconfiguring between PLL and other Transceiver Configuration Rules.

After a transceiver channel is used in PLL mode, it cannot be used for a usual transceiver operation. You have to connect the output of the Native PHY IP core in PLL mode to the respective transceiver input.

Figure 42. E-Tile Native PHY IP PLL Mode

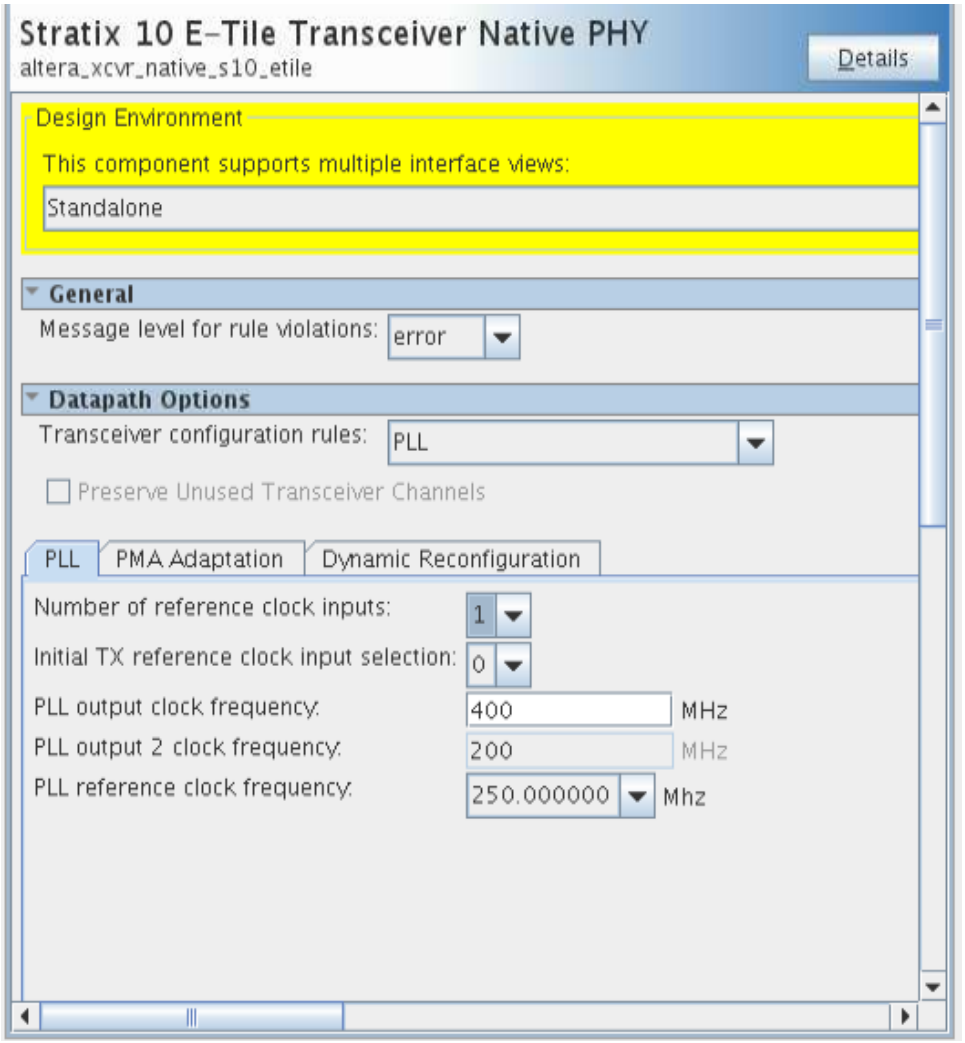


Table 31. E-Tile Native PHY IP PLL Mode Options

Parameter	Value	Description
Number of reference clock inputs	1, 2, 3, 4, 5	Specifies the desired number of reference clocks. The Native PHY IP core presents up to five clock inputs.
Initial TX reference clock input selection	Based on the number of reference clock input	Specifies the initially selected PLL reference clock input. This indicates the starting clock input selection used for this configuration when dynamically switching between multiple TX reference clock inputs.
PLL output clock frequency	100 - 1000 MHz	Specifies the PLL output frequency in units of MHz.
PLL output 2 clock frequency	Output from the PLL. `pll_clkout2` frequency = 0.5 * `pll_clkout1` frequency	Specifies the PLL output frequency for output 2 in units of MHz.
PLL reference clock frequency	Refer to the *Device Data Sheet*.	Selects the reference clock frequency for the PLL.

Table 32. E-Tile Native PHY IP PLL Ports

Port Name	Direction	Width	Description

`pll_refclk0`
`pll_refclk1`
`pll_refclk2`
`pll_refclk3`
`pll_refclk4`
	Input	1 bit for each channel	Reference clock for the PLL.

`pll_locked`	Output	1 bit for each channel	Locked status signal of the PLL. When you use the PLL channel as a clock source to the transceiver channel through external EMIB clocking: • Wait until `pll_locked` from the PLL channel is asserted before deasserting the transceiver channel reset at power-up. • If `pll_locked` from the PLL channel is deasserted at any time, hold the respective transceiver channels in reset until `pll_locked` is reasserted.
`tx_serial_data`	Output	1 bit for each channel	The `tx_serial_data` port is only used to assign the location of the PLL channel. It is not an active port for transmitting data.
`pll_clkout1` `pll_clkout2`	Output	1 bit for each channel	Output from the PLL. `pll_clkout2` frequency = 0.5 * `pll_clkout1` frequency

Related Information

- Four 25 Gbps PMA Direct Channel (with FEC) within a Single FEC Block on page 117
- Intel Stratix 10 Device Data Sheet
- Intel Agilex Device Data Sheet

2.2.12. Simplex Support

The Native PHY IP has three options for **Transceiver mode**: **TX/RX Duplex**, **TX Simplex**, and **RX Simplex**.

The simplex mode for the E-tile Native PHY IP permits one or more E-tile transceiver channels to be configured for transmit only or receive only. The unused portion of the channels (receive or transmit) is powered down to the extent permitted by the E-tile hardware operation. For some applications, only the transmitter or receiver of the high-speed serial transceiver is used. In these applications, it is desirable to power down the unused portion of the transceiver to save power.

The simplex mode applies only to **PMA direct** and **PMA direct high data rate PAM4**. RS-FEC modes and gearbox modes are not supported in simplex operation.

The interfaces to the E-tile Native PHY in simplex mode are essentially the same as in duplex mode. The unused side of the parallel data interface is terminated in simplex mode.

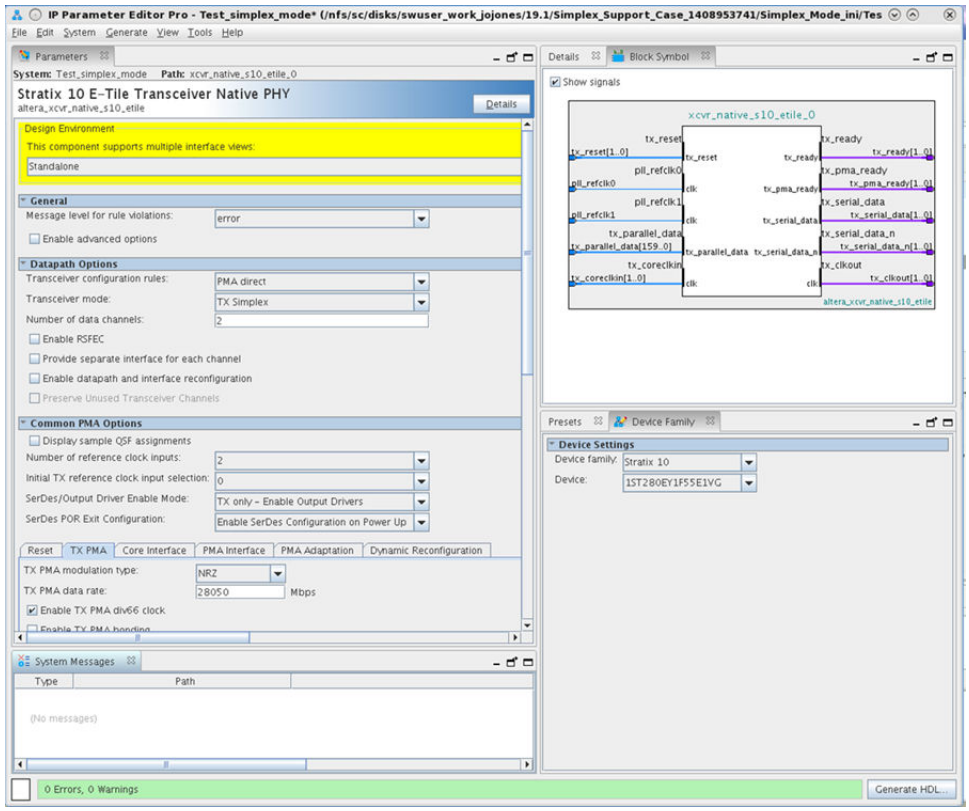
TX Simplex Mode

Select **Transceiver mode ➤ TX Simplex** in the Native PHY IP GUI as shown below.

When **TX Simplex** is selected:

- Set **Transceiver configuration rules** to **PMA direct** or **PMA direct high data rate PAM4**.
- The RS-FEC block must be turned off.
- Set the **SerDes/Output Driver Enable Mode** to one of the valid TX only selections. In the figure below, output drivers are enabled, and the PMA is set to TX only mode. The output drivers may also be disabled. This is useful for parallel loopback testing.

Figure 43. TX Simplex Mode



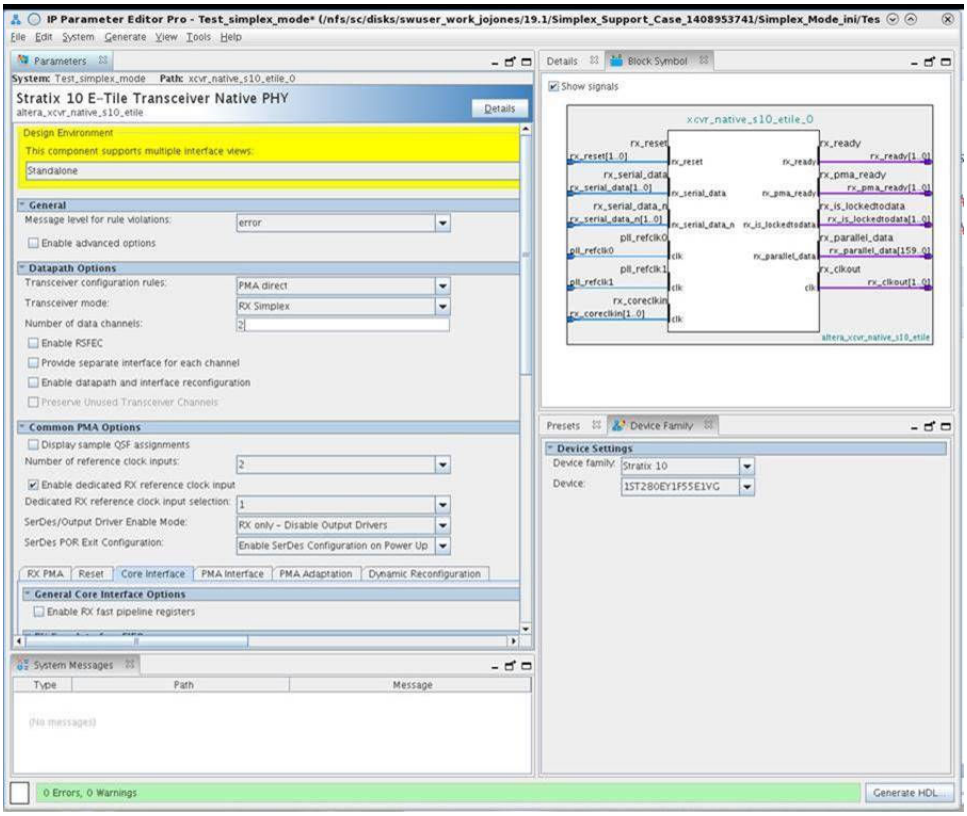
RX Simplex Mode

Select **Transceiver mode ➤ RX Simplex** in the Native PHY IP GUI as shown below.

When **RX Simplex** is selected:

- Set **Transceiver configuration rules** to **PMA direct** or **PMA direct high data rate PAM4**.
- The RS-FEC block must be turned off.
- Set **SerDes/Output Driver Enable Mode** to **RX only - Disable Output Drivers**.
- The IP parameter `duplex_mode` is set. The valid values of this parameter are displayed in the control. The validation procedure that sets the valid parameters is `::altera_xcvr_native_s10_etile::parameters::validate_duplex_mode`.

Figure 44. RX Simplex Mode



Simplex Support Parameters

Below are the specific parameters that must be set. If invalid values are supplied for these non-derived parameters, the hardware Tcl framework generates an error when the IP generation runs.

Table 33. Simplex Support Parameters

Parameter	Allowed Values	Comments
Transceiver configuration rules	`pma_direct` `pma_direct_high data rate PAM4`	Specifies **PMA direct** or **PMA direct high data rate PAM4** mode.
Transceiver mode	`tx` `rx`	Select `tx` for TX simplex mode and `rx` for RX simplex mode.
Enable RSFEC	`0`	Only `0` is valid for simplex modes.

SerDes/Output Driver Enable Mode	`Enable Output Drivers for TX Simplex`
	`Disable Output Drivers for TX Simplex`
	`Disable Output Drivers for RX Simplex`
		In TX simplex mode, the output drivers may be either enabled or disabled. In RX simplex mode, the output drivers must be disabled.

2.3. Implementing the Transceiver PHY Layer Revision History

Document Version	Changes
2020.01.31	Made the following changes: • Added **Enable de-skew**, **Preserve Unused Transceiver Channels**, **SerDes/Output Driver Enable Mode**, and **SerDes POR Exit Configuration** options to *General and Datapath Parameters*. • Added two figures: "Deskew Pulse with Double Width Mode Off (Full-Rate)" and "Deskew Pulse with Double Width Mode On (Half-Rate)."
2019.10.11	Made the following changes: • Updated *RX Deskew Logic*. • Added "Bit Mapping for Native PHY TX and RX Datapaths." • Added *Simplex Mode*. • Added the Related Information links for the Intel Agilex device documents.
2019.07.29	Made the following changes: • Added this clarification to *PLL Mode*: It is used for external EMIB clocking configurations (see the use case in *Four 25 Gbps PMA Direct Channel (with FEC) within a Single FEC Block*. • Added value descriptions for the "RS-FEC 78 Bit Bus" and "80 Bit Data Native PHY Double-Width TX/RX Ports" tables. • Added the "E-Tile Native PHY PLL Ports" table. • Added the *Deskew Logic*. • Added a link to the "Latency Measurement" section of the *E-tile Hard IP User Guide: E-tile Hard IP for Ethernet and E-Tile CPRI PHY Intel FPGA IPs* for the latency ports.
2019.04.19	Made the following changes: • Updated the "TX PMA Pre-equalization" table to agree with the "TX Equalization Settings for PAM4 and NRZ Signals" table. • Added `rsfec_signal_ok` and `i_rsfec_pld_ready` ports to the "Port Information" table. • Added *Gearbox 64/66*. • Added "Bit Mapping for Native PHY TX and RX Datapaths."
2019.02.04	Made the following changes: • Added *PMA Adaptation*. • Changed the maximum TX PMA reference clock frequency from 500 to 700. • Changed the maximum RX PMA reference clock frequency from 500 to 700.
2018.10.08	Made the following changes: • Changed the description of the design flow in the "Transceiver Design Flow in the Native PHY IP Core" section. • Added new parameters and updated parameter values and descriptions in the "General, Datapath Options, and Common PMA Options" table. • Added a note to the description of the **TX PMA data rate** parameter in the "TX PMA Options" table. • Added the following sections: — Reed Solomon Forward Error Correction (RS-FEC) Parameters — Fibre-channel and CPRI Modes — 128 GFC Mode — 25 GbE FEC Direct Mode — Interlaken Mode • Changed the description of the **Use default TX PMA pre-equalization settings** parameter in the "TX PMA Pre-equlization" table. • Changed the "PMA Interface Options" figure.

continued...

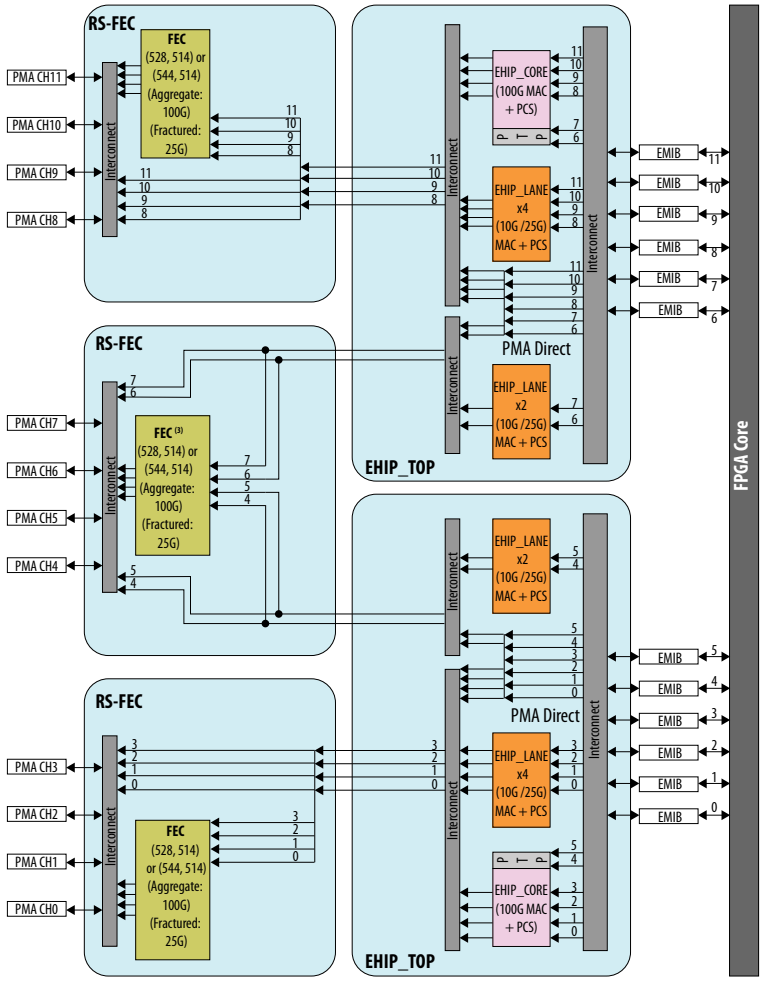
Document Version	Changes
	• Updated the "Native PHY IP Core Parameter Editor" figure. • Updated the "General, Datapath, and Common PMA Options" figure. • Added the following parameters to the "PMA Interface Options" table: — **Enable tx_enh_pmaif_fifo_almost_full port** — **Enable tx_enh_pmaif_fifo_almost_empty port** — **Enable tx_enh_pmaif_fifo_overflow port** — **Enable tx_enh_pmaif_fifo_underflow port** — **Enable rx_pmaif_fifo_underflow port** — **Enable rx_enh_pmaif_fifo_overflow port**
2018.08.08	Made the following changes: • Changed the description of the **Transceiver mode** parameter in the "General, Datapath, and Common PMA Options" table.
2018.07.18	Made the following changes: • Added further description about deskew bits to the PMA Direct high data rate PAM4 mode in the "Parallel Data" table. • Removed the "Port Diagram" figure.
2018.05.15	Made the following changes: • Added further description of the PMA Direct modes in the "Transceiver Design Flow in the Native PHY IP Core" section. • Updated the "Native PHY IP Core Parameter Editor" figure. • Added more description to the following parameters in the "Core Interface Parameters" table: — **Enable TX double width transfer** — **Enable RX double width transfer** • Added the "Port Information" table. • Changed the following parameters in the "General, Datapath Options, and Common PMA Options" table: — Removed the **Enable RS-FEC** parameter — Removed the **Enable datapath and interface reconfiguration** parameter — Changed the values and description for the **Transceiver mode** parameter • Added the following parameters to the "TX PMA Options" table: — **TX PMA clockout post divider** — **TX PMA reference clock frequency** • Added the following parameters to the "RX PMA Options" table: — **RX PMA clockout post divider** — **RX PMA reference clock frequency** • Added the "Reset Parameters" section. • Added the "Dynamic Reconfiguration Parameters" section. • Added the following ports to the "Port Information" table: — `reconfig_waitrequest` — `reconfig_readdata` — `reconfig_writedata` — `reconfig_address` — `reconfig_read` — `reconfig_write` — `reconfig_reset` — `reconfig_clk` • Added the "Parallel Data" table.
2018.01.31	Initial release.

3. E-Tile Transceiver PHY Architecture

Shown below are all the possible connections of the E-Tile Architecture. The datapath enablement depends on the configuration you are implementing. Refer to the E-Tile Channel Placement Tool for possible configurations.

Figure 45. E-Tile Architecture

Showing 12 out of 24 channels per tile.



Legend:

= EHIP_CORE = FEC = EHIP_LANE

Notes:

1. Not all datapath combinations are available.
2. Datapath enablement depends on the configuration you are implementing. Refer to the E-Tile Channel Placement tool for possible configurations.
3. This FEC block can only be used in aggregate mode with FEC direct application (e.g. 128GFC Fibre-Channel). This FEC block cannot be used in in aggregate mode with EHIP_CORE because there is no EHIP_CORE in this location.

Related Information

E-Tile Channel Placement Tool

3.1. Physical Medium Attachment (PMA) Architecture

The PMA acts as the analog front end for the E-tile transceivers.

The PMA transmits and receives high-speed serial data depending on the transceiver channel configuration. The PMA transmitter serializes parallel data, and the PMA receiver deserializes serial data.

The E-tile PMA GXE channels support both NRZ and PAM4 data formats. A single bit of data is transmitted/received in one UI in NRZ mode, while two bits of data are transmitted/received in one UI in PAM4 mode. The transceiver can operate up to 28.9 Gbps in NRZ mode and 57.8 Gbps in PAM4 mode.

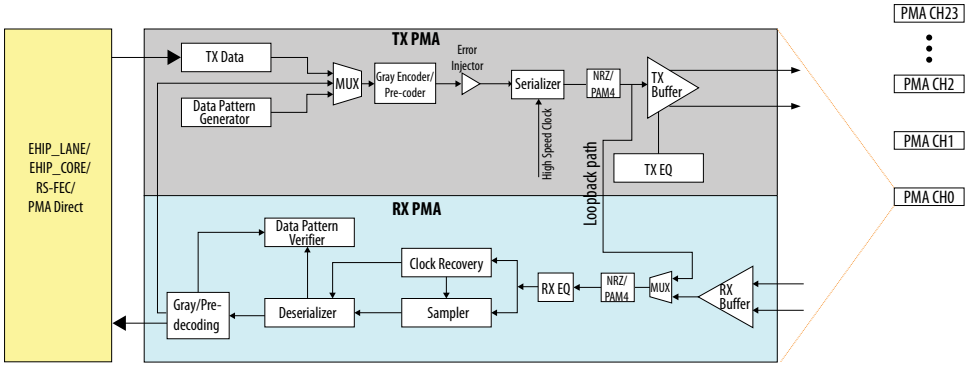
The PMA supports the following parallel data widths:

- 16 bits (NRZ mode only)
- 20 bits (NRZ mode only)
- 32 bits (NRZ, PAM4)
- 40 bits (NRZ, PAM4)
- 64 bits (PAM4 high data rate mode only)

Supported protocols include, but are not limited to:

- IEEE 802.3ap (10GBASE-KR)
- IEEE 802.3bj (100G-KR4, 100G-CR4)
- IEEE 802.3bm (CAUI4)
- IEEE 802.3bs (400G Ethernet)
- IEEE 802.3cd (50GBASE-CR, 50GBASE-KR)
- IEEE 802.3by (25GBASE-CR, 25GBASE-KR)
- CEI-25G-LR
- CEI-28G-VSR/SR/MR
- CEI-56G-VSR/MR/LR
- 32GFC

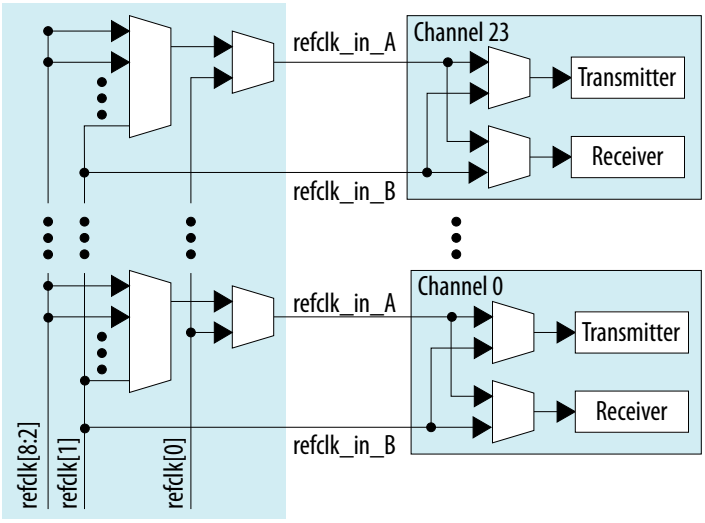
Figure 46. PMA Architecture Block Diagram



A given E-tile has nine reference clock pins linked to a reference clock network, which is shared across all of the 24 PMA channels within a tile. `refclk_0` routing is skew-balanced across all the channels and is used for TX PMA bonding. Additionally, each channel has two clock input ports (`refclk_in_A` and `refclk_in_B`) which drive

dedicated clocking resources. Muxing options allow you to select the desired external reference clock pin to drive the individual clock input ports for each PMA channel. The block diagram below demonstrates the muxing capability.

Figure 47. Dynamically Selected Reference Clocks



For more details, refer to *Clock Network*.

3.1.1. Transmitter PMA

The transmitter data path of the PMA comprises of a transmitter buffer (TX Buffer), transmitter equalizer (TX EQ), and a serializer driven by a clock network dedicated to each transceiver channel.

Parallel data received from the EHIP_LANE, EHIP_CORE, RS-FEC, and PMA Direct is serialized by the serializer. This serialized data is de-emphasized to compensate for the intersymbol interference (ISI) losses using an equalizer. The transmitter buffer at the end of the PMA data path shapes the signal and drives the serialized data off the chip.

3.1.1.1. High Speed Differential Transmitter

The transmitter buffer includes the following circuitry:

- High speed transmitter line buffer
- Transmitter equalizer

3.1.1.1.1. High Speed Transmitter Line Buffer

The transmitter differential I/O buffer converts the serialized bit stream to an electrical signal suitable for transmission across a cable or PCB channel. The Attenuation Value (VOD) parameter controls the transmitter swing strength, and pre-tap/post-tap cursors help shape the transmitter output waveform.

On power-up reset and FPGA device configuration, the transceiver supply voltage is driven on the TX lines. During this state, both the transmitter and TX buffer are disabled. The TX buffer is in tristate during the start-up sequence. When both the

transmitter and TX buffer are enabled, the TX buffer drives normal differential data, and the differential impedance on both TX and RX lines is in the range of 80 (minimum), 100 (typical), 120 (maximum) Ω. For more details on the termination modes, refer to the description for PMA attribute code 0x002B in the *PMA Attribute Codes* section.

Note: Differential impedance values are not programmable (after power-up, the differential impedance typically ranges from 80 Ω to 120 Ω).

Related Information

- PMA Register Map on page 211
- PMA Attribute Codes on page 216

3.1.1.1.2. TX Equalizer

The TX equalizer has one Post-tap, three Pre-taps and one Main Tap (attenuation). The frequency response of the filter is chosen to compensate for the channel impairments such as intersymbol interference (ISI), crosstalk, frequency-dependent losses and reflections. When a signal is transmitted over a channel, the high-frequency components get attenuated. To save power, de-emphasis is used instead of pre-emphasis. This results in a bit stream which is pre-distorted by the equalizer over several bits.

Table 34. PMA Transmitter Programmable Parameters

Parameter	Mode
Attenuation (ATTN)	PAM4, NRZ
Post-tap1 (POST)	PAM4, NRZ
Pre-tap1 (PRE1)	PAM4, NRZ
Pre-tap2 (PRE2)	PAM4, NRZ
Pre-tap3 (PRE3)	PAM4, NRZ

Refer to the resources linked below to help calculate transmitter equalization.

Details on TX equalization settings for PAM4 and NRZ signals

TX equalization parameters take the following range.

Table 35. TX Equalization Settings for PAM4 and NRZ Signals

Cursor	Rule	Step Size
ATTN	0 ≤ ATTN ≤ 26	18.5 mV/step (0.17 dB)
POST	-18 ≤ POST ≤ 18 Increment/decrement by 2	18.5 mV/step (0.34 dB)
PRE1	-10 ≤ PRE1 ≤ 10 Increment/decrement by 2	18.5 mV/step (0.34 dB)
PRE2	-15 ≤ PRE2 ≤ 15	9.25 mV/step (0.17 dB)
PRE3	-1 ≤ PRE3 ≤ 1	9.25 mV/step (0.17 dB)

Note: Where ABS is the absolute value, ABS(PRE3) + ABS(PRE2) + ABS(PRE1) + ATTN + ABS(POST) ≤ 32.

For more details on TX equalization programmability, refer to the description for code 0x0015 in the *PMA Attribute Codes* section.

The ATTN/PRE/POST taps can be dynamically changed via the Avalon-MM interface.

For more details on PMA attribute support and programming, refer to *PMA/PCS Avalon-MM Register Map* and *PMA Attribute Codes* to configure these parameters.

Related Information

- PMA Register Map on page 211
- PMA Attribute Codes on page 216
- E-Tile Transmitter Equalization Tool
- PAM4—Transmitter Equalization Table
- NRZ—Transmitter Equalization Table
  These tables are intended to look up a given combination of transmitter equalization settings. The Input columns are the input values you enter, and the Outputs columns are the actual output values after programming. You can look up whether your input-output combination is valid or invalid and whether your inputs match your outputs.

3.1.1.2. Gray Encoder/Precoder

PAM4 patterns generated in PAM4 mode are gray encoded by default. The encoding is as follows:

Table 36. Gray Encoding

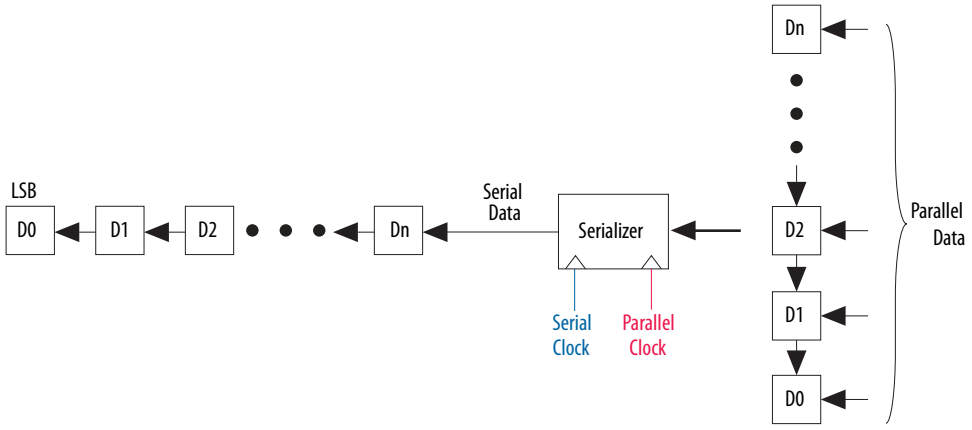
Linear	Gray
00	00
01	01
10	11
11	10

The transmitter also includes a precoder that you can optionally 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.

3.1.1.3. Serializer

The serializer converts the received parallel data into a serial data stream. The channel serializer supports the following serialization factors: 16, 20, 32, 40, and 64. The serializer is hard-coded to LSB first (in both the TX and RX directions).

Figure 48. Serializer



3.1.1.4. Data Pattern Generation

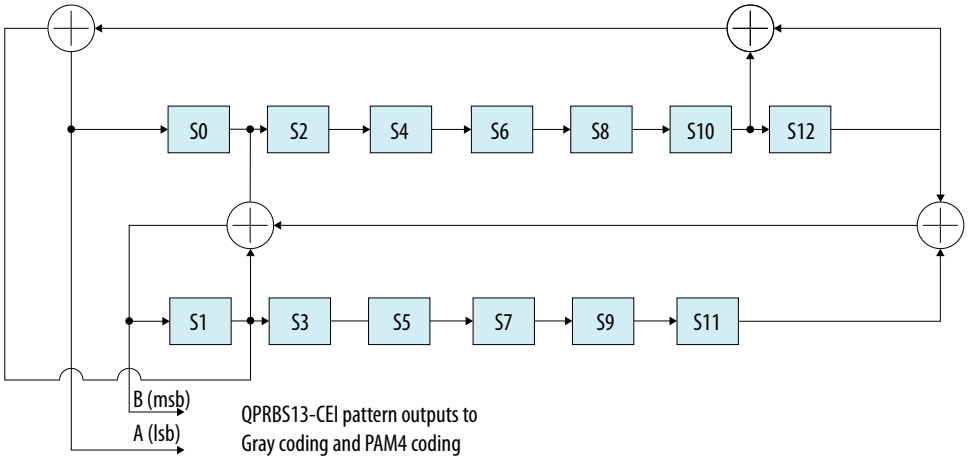
The data pattern generator is a Design for Test (DFT) feature capable of generating data traffic for the PHY to debug the PMA without involving the upper protocol stack layers.

The E-tile has an on-chip pseudo random pattern generation block that operates in all bit modes and can generate several patterns. In addition to this, it can generate an 80-bit user-defined pattern.

There are patterns supporting both NRZ and PAM4. Pseudo random bit sequence (PRBS) NRZ patterns are different from PAM4 patterns. Different specifications such as CEI OIF and IEEE 803.2, refer to quaternary PAM4 patterns differently. QPRBS13 is identical to PRBSQ13 and QPRBS31 is identical to PRBSQ31.

As defined in OIF Clause 16: CEI-56G-VSR-PAM4 Very Short Reach Interface specifications, typically, each cycle of PRBSQ13 is 8191 unique symbols long. Each cycle is formed by gray coding and PAM4 encoding of bits from two repetitions of the PRBS13 pattern and used for transmitter compliance testing.

Figure 49. QPRBS13-CEI Pattern



For more details on PRBS13Q pattern generation, refer to CEI-56G-VSR-PAM4 specifications. On similar lines, the PRBS31Q pattern is a repeating 2^{31} -1 symbols long, formed by gray coding and PAM4 encoding of the PRBS31 pattern. This pattern is used for receiver testing.

Note: Gray encoding is enabled by default in PAM4 patterns.

Table 37. Supported Programmable NRZ and PAM4 Patterns

NRZ Mode	PAM4 Mode
PRBS7	PRBS7Q
PRBS9	PRBS9Q
PRBS11	PRBS11Q
PRBS13	PRBS13Q
PRBS15	PRBS15Q
PRBS23	PRBS23Q
PRBS31	PRBS31Q
User-defined 80-bit Pattern	

More details on these patterns are as follows:

2^7–1 PRBS pattern This standard PRBS pattern [8] is based on the generator polynomial $x^7 + x^6 + 1$ (refer to ITU V.29)

2^9–1 PRBS pattern This PRBS pattern is based on the generator polynomial $x^9 + x^5 + 1$ (refer to CCITT O.151/ITU-T O.151)

2^{11}–1 PRBS pattern This PRBS pattern is based on the generator polynomial $x^{11} + x^9 + 1$ (refer to CCITT O.151/ITU-T O.151)

2^{13}–1 PRBS pattern This PRBS pattern is based on the generator polynomial $x^{13} + x^{12} + x^2 + x + 1$ (refer to CCITT O.151/ITU-T O.151)

2^{15}–1 PRBS pattern This PRBS pattern [9] is based on the generator polynomial $x^{15} + x^{14} + 1$ (refer to CCITT O.151/ITU-T O.151)

2^{23}–1 PRBS pattern This PRBS pattern [10]is based on the generator polynomial $x^{23} + x^{18} + 1$ (refer to CCITT O.151/ITU-T O.151)

---

(8) This pattern repeats every 127 bits and you can use it with a PRBS receiver to facilitate loopback testing. This pattern facilitates the testing of chip-to-chip communications with other transceiver channel TX/RX Macro receivers on external chips or be fed to instruments such as a bit error rate tester (BERT).

(9) This polynomial provides a data pattern that is more challenging for clock and data recovery circuits. Run lengths up to 15 1s or 14 0s in a row are embedded in the pattern. The pattern repeats every 2^{15}–1 bits (approximately 32.8 Kb).

(10) This polynomial provides a data pattern that is more challenging for clock and data recovery circuits. Run lengths up to 23 1s or 22 0s in a row are embedded in the pattern. The pattern repeats every 2^{23}–1 bits (approximately 8.4 Mbits).

2^{31}–1 PRBS pattern This PRBS pattern [(11)]is based on the generator polynomial $x^{31} + x^{28} + 1$

For more details on Register Read/Write support and programming, refer to *PMA Register Map* and *PMA Attribute Codes* to configure these parameters.

Related Information

- PMA Register Map on page 211
- PMA Attribute Codes on page 216

3.1.2. Receiver PMA

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 EHIP_LANE, EHIP_CORE, RS-FEC, or the FPGA core.

The receiver portion of the PMA consists of the receiver buffer, the clock data recovery (CDR) unit, and the deserializer.

3.1.2.1. Receiver Buffer

The receiver buffer receives serial data from the input pins and feeds it to the clock data recovery (CDR) unit and deserializer.

The receiver buffer supports the following features:

- Programmable termination mode
- Receiver equalization

For more details on Register Read/Write support and programming, refer to *PMA Register Map* and *PMA Attribute Codes* to configure these parameters.

Related Information

- PMA Register Map on page 211
- PMA Attribute Codes on page 216

3.1.2.1.1. Programmable Termination Modes

Termination modes are programmable. However, the differential impedance values are fixed (as per the Ethernet standard specifications).

The transceiver RX is AC-coupled on-chip. Therefore, no off-chip AC-coupling capacitor is required provided that the RX input common mode is between AGND and VCCH_GXE and the RX input amplitude is < 1200 mVp-p differential. For details, refer the *Device Family Pin Connection Guidelines*. If the above requirements are not met, a typical value of 100 nF of capacitive termination can be used on the board.

For more details on Register Read/Write support and programming, refer to *PMA Register Map* and *PMA Attribute Codes* to configure these parameters.

(11) This polynomial generates data patterns whose run lengths are up to 31 1s or 30 0s in a row. The pattern repeats every 2^{31}–1 bits (approximately 2.15 Gbits).

Related Information

- PMA Register Map on page 211
- PMA Attribute Codes on page 216
- Intel Stratix 10 Device Family Pin Connection Guidelines
- Intel Agilex Device Family Pin Connection Guidelines

3.1.2.1.2. RX Adaptation Modes

The E-tile supports the initial and continuous adaptation modes.

Table 38. E-Tile Receiver PMA RX Adaptation Modes

Mode	Description
Initial adaptation	During initial adaptation, the adaptation engine adapts all of the RX analog front end (AFE) parameters to optimize the receiver eye opening and adjusts the vertical and horizontal sampling location accordingly. • This mode calibrates the PMA to known good settings. • It is disruptive tuning and impacts the data traffic. Run initial adaptation under either of the following conditions: • On device power up. Refer to *PMA Bring Up Flow* for more detail. • When there is change in the physical channel between TX and RX. • During debug, when you switch back and forth between internal serial loopback and mission mode and the BER readout is exceptionally high, Intel recommends that you issue a PMA reset followed by loopback mode and initial adaptation.
Continuous adaptation	The goal of this adaptation is to maintain the signal quality at the sampler close to the initial adaptation over time and temperature. • This mode is run only after running initial adaptation during the device bring up. • This mode tracks the temperature over time by continuously adapting new values of the RX AFE parameters. • This mode is a continuous and non-disruptive process, that is, it does not impact the data traffic. • During a link debug process with the hard PRBS generator and verifier, you cannot read out accumulated errors from the error counter unless you stop continuous adaptation. Details on PMA code and value to stop continuous adaptation are available in the *PMA Register Map*.

For more details on Register Read/Write support and programming, refer to *PMA Receiver Equalization Adaptation Usage Model* and *0x000A: Receiver Tuning Controls* to configure these parameters.

The following table lists all the PMA parameters that you can manually optimize before triggering initial adaptation and continuous adaptation. Refer to *PMA AVMM Registers* and *PMA Receiver Equalization Adaptation Usage Model* to understand how the parameter can be fixed such that it is not overwritten by the adaptation engine. If you do not fix the parameter, the adaptation engine adapts the parameter to the required value.

Table 39. PMA Parameter Description and Range

Some of these parameters are tunable during initial adaptation and continuous adaptation. To avoid getting an optimized parameter overwritten by the adaptation tuning engine, you must fix the parameter.

Parameter	Min	Max	Initial Adaptation	Continuous Adaptation	Manual Optimization Possible	Firmware Default
GainLF	0	15	Yes	Yes	Yes	8
CTLE LF Min	0	15	N/A	N/A	N/A	0

continued...

Parameter	Min	Max	Initial Adaptation	Continuous Adaptation	Manual Optimization Possible	Firmware Default
CTLE LF Max	0	15	N/A	N/A	N/A	15
GainHF	0	15	Yes	Yes	Yes	0
CTLE HF min	0	15	N/A	N/A	N/A	0
CTLE HF max	0	15	N/A	N/A	N/A	15
GS1	0	3	No	No	Yes	0
GS2	0	3	No	No	Yes	0
RF_P2	-10(12)	10	Yes	No	No	0
RF_P2_MIN	-10(12)	10	N/A	N/A	N/A	-10
RF_P2_MAX	-10(12)	10	N/A	N/A	N/A	10
RF_P1	0	15	Yes	Yes	No	0
RF_P1_MIN	0	15	N/A	N/A	N/A	0
RF_P1_MAX	0	15	N/A	N/A	N/A	15
RF_P0	-15(12)	15	Yes	Yes	No	0
RF_B1	0	8	Yes	Yes	Yes	0
RF_B0	0	5	No	Yes	Yes	0
RF_B0T	—	—	—	—	—	—
RF_A - NRZ	—	—	—	—	Yes	160
RF_A - PAM4	—	—	—	—	Yes	130

PMA Initial Adaptation Effort Status

- 0 = Low Effort (00_effort) is for NRZ Ethernet AN/LT and CPRI protocols only and is the quickest to complete. This meets the 500 ms compliance time for Ethernet and 100 ms compliance time for CPRI.
- 1 = Medium Effort (05_effort) is for the PAM4 Ethernet AN/LT protocol only, to meet the IEEE link up time of 3 s.
- 2 = Full Effort (10_effort) is for general usage (NRZ and PAM4), to provide the best performance and stability, but takes the most time to complete compared to other initial adaptation efforts. This is the recommended adaptation mode.

See the "Loading PMA Configuration Register START_ADAPTATION" figure.

You can set the initial adaptation effort using attributes directly. See *Initial Adaptation Effort Levels* for information on the attributes to send.

Related Information

- PMA Receiver Equalization Adaptation Usage Model on page 188
- 0x000A: Receiver Tuning Controls on page 221
- PMA Bring Up Flow on page 83
- PMA Register Map on page 211

(12) Two's complement, 16-bits

- Initial Adaptation Effort Levels on page 235
- PMA Registers 0x200 to 0x203 Usage on page 235
  See the "Loading PMA Configuration Register START_ADAPTATION" figure.

3.1.2.2. Clock Data Recovery (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.

3.1.2.3. Input Sampler

The Input Sampler block is responsible for converting the serial input signal into a retimed bit stream using the high-speed serial clock generated by the CDR block.

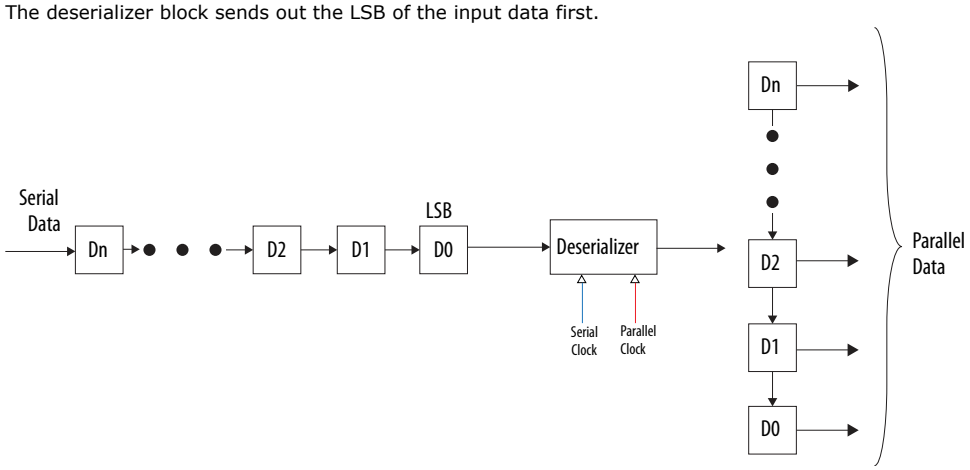
3.1.2.4. Deserializer

The deserializer block 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 channel deserializer supports the following deserialization factors: 16, 20, 32, 40, and 64.

Figure 50. Deserializer

The deserializer block sends out the LSB of the input data first.



3.1.2.5. Data Pattern Verifier

The data pattern verifier is used to verify the signal received at the receiver. The deserialized data pattern is sent to the data pattern verifier, which compares the received data pattern to the pattern it is configured to. There are several patterns which can be verified in NRZ and PAM4 mode. For the same setting, depending on the encoding mode, either PRBSx (NRZ) or PRBSxQ (PAM4) is configured.

Table 40. PRBS Patterns by Mode

NRZ Mode	PAM4 Mode
PRBS7	PRBS7Q
PRBS9	PRBS9Q
PRBS11	PRBS11Q
PRBS13	PRBS13Q
PRBS15	PRBS15Q
PRBS23	PRBS23Q
PRBS31	PRBS31Q
User-defined pattern 80-bit	

3.1.3. PMA Tuning

The E-tile PMA supports various data rates and channel configurations to meet the most advanced PAM4 and NRZ protocols across a temperature range. The default adaptation sequence works at a static temperature regardless of the load.

For certain protocols, the adaptation sequence may not yield optimal performance over the temperature sweep. When temperature increases, the signal quality at the sampler degrades and the vertical and horizontal eye opening at the sampler degrade. To compensate for these degradations, RX AFE parameters and decision feedback equalization (DFE) must be positioned so that the adaptation results in an optimal performance with environmental changes.

PMA tuning methodology optimizes link performance to reduce the bit error rate (BER).

3.1.3.1. Purpose of PMA Tuning

You can see the value of PMA tuning by considering the following two cases:

- Case 1—Optimum link performance measured by executing initial adaptation at static (minimum and maximum) temperatures
- Case 2—Optimum link performance for a temperature sweep (real time dynamic temperature condition) may not be achieved with initial adaptation and continuous adaptation.

Figure 51. Case 1: Static Temperature

When initial adaptation is initiated at both low and high temperature (static temperature cases) it results in relatively low BER at both the temperatures.

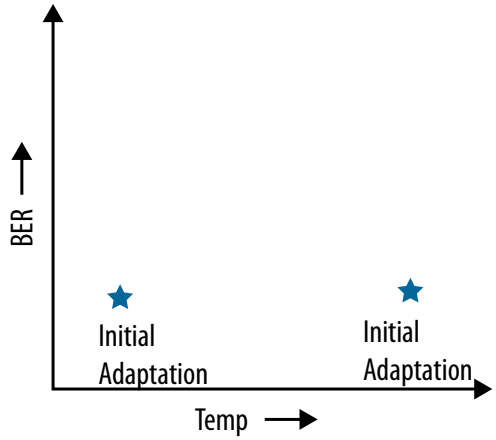
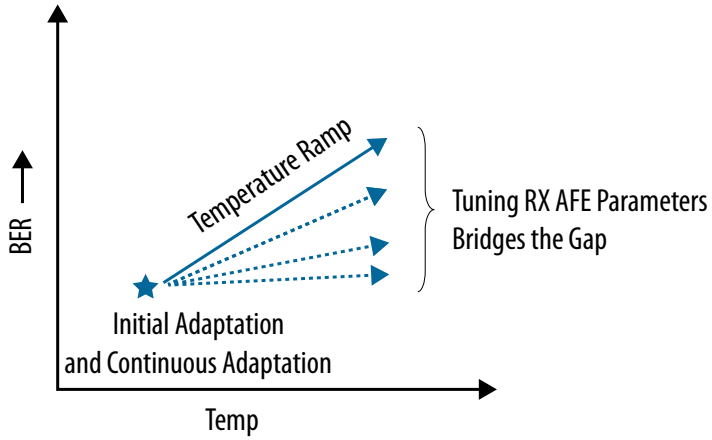


Figure 52. Case 2: Dynamic Temperature Ramp

The link is brought up with initial adaptation at a static temperature (low).



The temperature is increased with continuous adaptation running in the background. Continuous adaptation is required in dynamic temperature conditions and its adaptive range is highly dependent on initial adaptation conditions. The goal of continuous adaptation is to compensate for temperature changes and ensure the link performance is as near as possible to the initial adaptation performed at the low temperature in Case 2. As the temperature increases, the eye degrades at the sampler and reports a relatively higher BER. Continuous adaptation started at a lower temperatures does not result in the optimal performance observed at higher static temperature in Case 1 after running initial adaptation. This implies that continuous adaptation is unable to maintain the performance obtained after running initial adaptation.

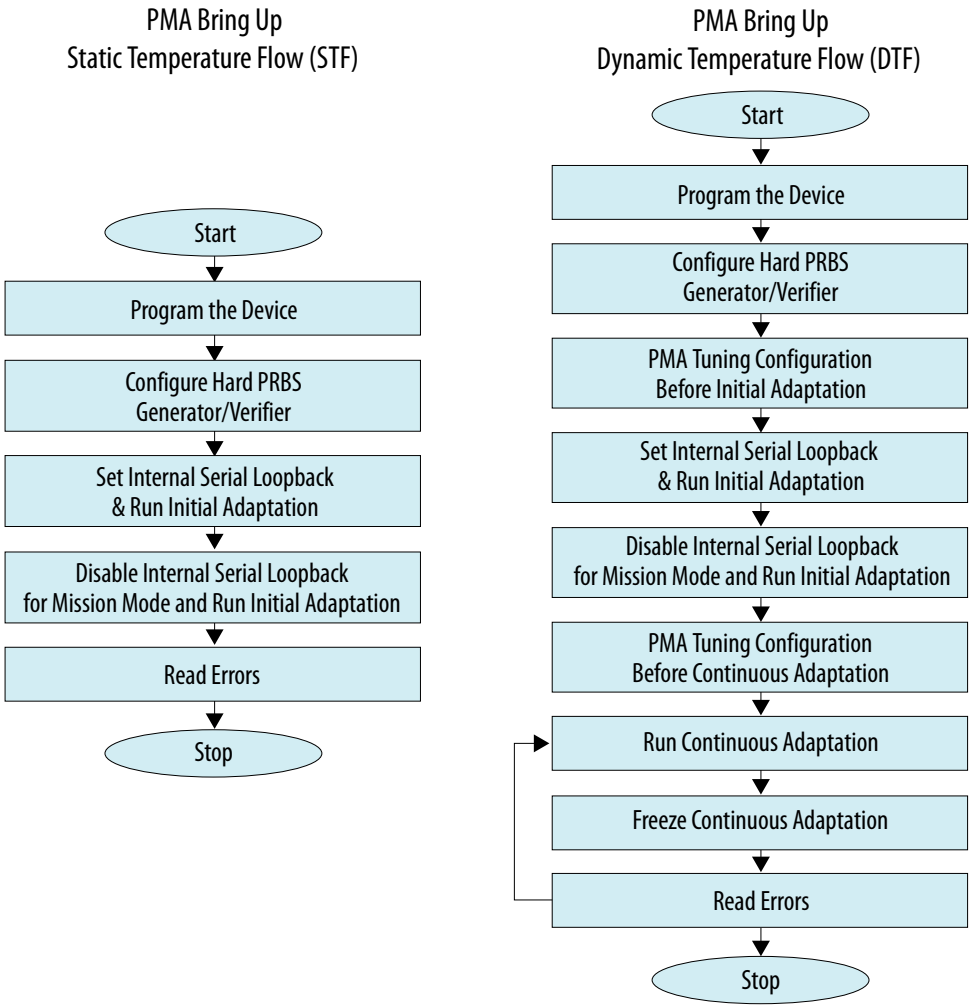
You can reduce this link performance degradation between the static temperature (Case 1) and dynamic temperature (Case 2) conditions by tuning the PMA AFE parameters. This establishes the desired dynamic temperature performance of continuous adaptation close to the initial adaptation at high temperature.

3.1.3.2. PMA Bring Up Flow

Static temperature flows (STF) and dynamic temperature flows (DTF) have different processes for successful link bring up.

The DTF link bring up flow is used in real time scenarios to compensate for dynamic temperature conditions. Unlike STF, DTF link bring up runs continuous adaptation to maintain link performance over the dynamic temperature conditions. Configure the PMA parameters before running initial adaptation and continuous adaptation to extend the adaptive range of PMA in DTF link bring up.

Figure 53. STF and DTF Link Bring Up



Note: In DTF and STF modes, initial adaptation is run in internal serial loopback mode to calibrate the AFE parameters.

Note: In DTF and STF modes, initial adaptation is run in mission mode to calibrate AFE parameters with regards to the connected ISI channel.

Note: Always run continuous adaptation after the following; performing the steps in this order is a must to get the PMA in the correct state:

- Internal serial loopback followed by initial adaptation
- Mission mode followed by initial adaptation

Note: When the device is in mission mode with continuous adaptation running and the ISI channel is changed, rerun initial adaptation. If the CDR lock does not assert or an unexpectedly high BER is recorded, perform a PMA analog reset. Refer to PMA reset for more details.

Note: Completion of initial adaptation can be read out by polling the PMA register. For more details refer to the *PMA Register Map*.

Note: During PMA performance verification testing, with continuous adaptation running in background, error bits cannot be accumulated to calculate BER because the Hard PRBS error counter is in a busy state. You can read errors during continuous adaptation by implementing a soft PRBS generator and verifier. Errors can be accumulated in hard PRBS error counter after stopping the continuous adaptation.

Note: When you generate and verify a pattern with logic, you can ignore the freeze continuous adaptation step in the dynamic temperature flow.

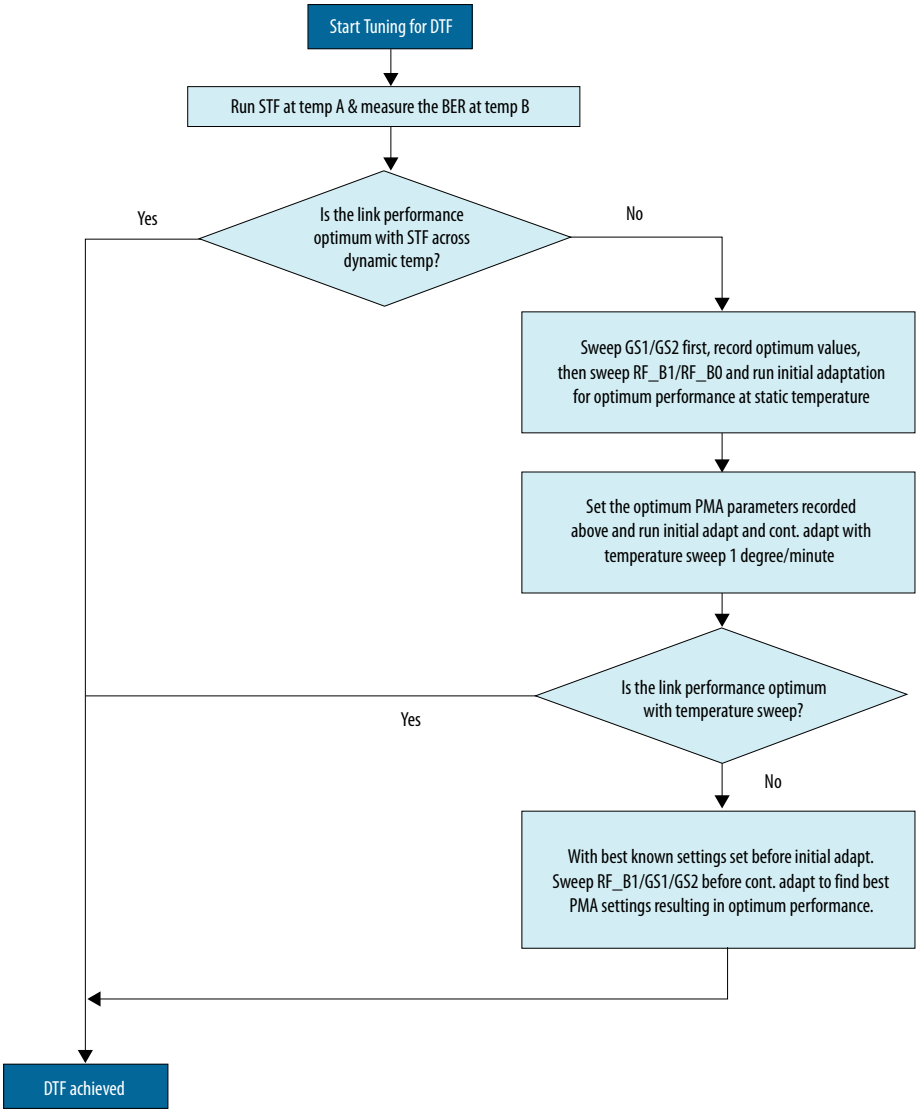
Related Information

- PMA Register Map on page 211
- PMA Analog Reset on page 132

3.1.3.3. PMA Tuning Guidelines

1. Run the STF across the desired temperature sweep. There is no need to run the DTF if the STF provides a good BER across the temperature sweep.
2. If optimal performance is not achieved, then sweep GS1 and GS2 first. Then, enter the best settings, sweep the RF_B0 and RF_B1 parameters, and run initial adaptation at static temperatures (low and high). After understanding the trend of these parameters at static temperature, record the optimum value of these parameters which results in the optimum performance across the desired temperature sweep range.
3. Set the optimum parameters you recorded and initiate initial adaptation. Sweep the temperature in steps of 1°C per minute and run continuous adaptation to track the temperature variations in PMA.
   This bring up flow becomes the DTF if the link performance is optimum across the temperature sweep.
4. If optimal performance is not achieved, then add a sweep of the RF_B1/GS1/GS2 parameters and initiate continuous adaptation to the bring up flow used in Step 3. When you have achieved optimal link performance, you have completed the DTF bring up.

Figure 54. PMA Tuning Generic Flow



You can use the PMA parameter configurations in the table below for the specified data rate and channel loss. Refer to Table 39 on page 78 for details about the parameters listed below.

Table 41. PMA Parameter Tuning for Extending Dynamic Range

These are the tuned PMA parameter settings that result in optimal link performance across a temperature sweep for each test configuration. An integer value means that the parameter is fixed, and **Firmware Default** means that the parameter is adaptive. Use the following PMA configurations as a starting point across process voltage and temperature. Further tuning is required if the BER does not meet the required protocol specifications.

Parameter	PAM4		NRZ		
	56 Gbps		28 Gbps		10 Gbps
	LR	VSR	LR	VSR	25 dB/15 dB/10 dB
PMA Tuning Configuration Before Running Initial Adaptation					
RF_A	130	130	130	160	160
GS1	1	0	2	0	2
GS2	1	0	2	0	1
RF_B0	2	3	1	Firmware Default(13)	1
RF_B1	8	3	1		5
CTLE LF maximum	2	15	3	15	15
CTLE LF minimum	0	0	0	7	0
CTLE HF maximum	15	15	15	7	15
CTLE HF minimum	0	0	0	0	0
RF_P1_MAX	6	6	6	15	15
RF_P2	Firmware Default(13)	Firmware Default(13)	0	Firmware Default(13)	0
RF_B0T	40	10	10	10	10
PMA Tuning Configuration Before Running Continuous Adaptation					
RF_B1	Set to adaptive		8	Set to adaptive	

3.1.3.4. General PMA Tuning Guidelines

Observe these guidelines for custom data rate and channel situations. Data rates of 6.5 Gbps and below can use manual CTLE if a long initial adaption time is not acceptable.

(13) Find the firmware default values in Table 39 on page 78.

- Minimize your use of TX equalization.

Table 42. Recommended TX Attenuation Value (VOD)

Insertion Loss	< 10 dB	> 10 dB
TX equalization Attenuation Value (VOD)	6-8 This is only useful for loopback tests; it does not apply for actual applications. For actual applications, tune the launch VOD of the link partner to get the optimal BER at the E-tile receiver.	default

- Sweep the GS1 and GS2 parameters when possible.

Table 43. Typical GS1/GS2 Settings

For lower data rates with higher insertion loss, you can use higher values.

Insertion Loss	< 13 dB	> 13 dB
GS1	0	1
GS2	1	2

- The typical values for RF_B0 and RF_B1 are 1 and 4. For high data rates (50 Gbps and above) and for longer channels (20 dB and above), you can use a RF_B1 value of 6 or 8 (maximum) to take advantage of full EQ capabilities. Depending on the system and temperature ramp, you may be required to use a RF_B0 setting of 2 to allow sufficient room on both up and down ramps.
- Sweep the RF_B0/RF_B1 values and apply optimized values for a channel or group (short/medium/long) of channels whenever possible.
- During continuous adaptation, fixing LF_Max (which limits the LF adaptation range) and adapting RF_B0 provides the best performance over temperature ramp.

3.1.4. Duplex Adaptation Flow

1. If independent TX and RX reset is disabled, assert then deassert the `reset` signal. Otherwise, assert then deassert the `tx_reset` and `rx_reset` signals. Wait for `tx_ready` and `rx_ready` to assert.
2. Trigger PMA analog reset(14).
3. Reload PMA settings (call the PMA attribute sequencer) using 0x91[0] = 1(14).
4. If using traffic, enable PRBS31. Otherwise, it is optional.
5. If using a PMA configuration, load the PMA configuration using control status registers (CSR). This is loaded to the registers using PMA registers 0x200 to 0x203(15).
   a. Write 0x40143 = 0x80.
   b. Read 0x40144[0] until it changes to 0.
   c. Load the PMA configuration using PMA registers 0x200 to 0x203.
6. Enable internal serial loopback(16).

(14) Refer to *PMA Analog Reset*.

(15) Refer to *Loading a PMA Configuration* and *PMA Registers 0x200 to 0x203 Usage*.

7. Run initial adaptation using PMA attributes[17].
8. Verify that the initial adaptation status is complete using PMA attribute code 0x0126 and data 0x0B00.
9. Enable mission mode and disable internal serial loopback (skip this step if using internal serial loopback)[16].
10. If not using PRBS, disable it.
11. If valid data rate traffic is available at the RX, proceed to the next step. Otherwise, rerun initial adaptation until valid traffic.
12. Run initial adaptation using PMA attributes[17] (skip this step if using internal serial loopback).
13. Verify that the initial adaptation status is complete using PMA attribute code 0x0126 and data 0x0B00 (skip this step if using internal serial loopback).
14. Run continuous adaptation[17].
15. If independent TX and RX reset is disabled, assert then deassert the `reset` signal. Otherwise, assert then deassert the `tx_reset` and `rx_reset` signals. Wait for `tx_ready` and `rx_ready` to assert.
16. Optional: Check the link status with `rx_is_lockedtodata` (clear the traffic checker, and verify that it is error free).
17. Start transmitting and receiving data.

Related Information

- PMA Analog Reset on page 132
- Loading a PMA Configuration on page 204
- 0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control on page 220
- 0x000A: Receiver Tuning Controls on page 221
- PMA Registers 0x200 to 0x203 Usage on page 235

3.1.5. RX Simplex Adaptation Flow

1. Assert the `rx_reset` signal.
2. Trigger PMA analog reset[18].
3. Reload PMA settings (call the PMA attribute sequencer) using 0x91[0] = 1[18].
4. If using a PMA configuration, load the PMA configuration using control status registers (CSR). This is loaded to the registers using PMA registers 0x200 to 0x203[19].
   a. Write 0x40143 = 0x80.

(16) For how to enable and disable internal serial loopback, refer to *0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control*.

(17) For PMA attributes for adaptation, refer to *0x000A: Receiver Tuning Controls*.

(18) Refer to *PMA Analog Reset*.

(19) Refer to *Loading a PMA Configuration* and *PMA Registers 0x200 to 0x203 Usage*.

b. Read 0x40144[0] until it changes to 0.
c. Load the PMA configuration using PMA registers 0x200 to 0x203.

5. Disable internal serial loopback[20].
6. Run Initial Adaptation using PMA attributes[21].
7. Verify that the initial adaptation status is complete using PMA attribute code 0x0126 and data 0x0B00.
8. If valid data rate traffic is available at the RX, proceed to the next step. Otherwise, rerun initial adaptation until valid traffic is available.
9. Run continuous adaptation[21].
10. Deassert the `rx_reset` signal.
11. Optional: Check the link status with `rx_is_lockedtodata` (clear the traffic checker, and verify that it is error free).
12. Start receiving data.

Related Information

- PMA Analog Reset on page 132
- Loading a PMA Configuration on page 204
- 0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control on page 220
- 0x000A: Receiver Tuning Controls on page 221
- PMA Registers 0x200 to 0x203 Usage on page 235

3.1.6. Dynamic Reconfiguration Adaptation Flow

Refer to *Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode* for a dynamic reconfiguration flow with PMA adaptation.

1. If independent TX and RX reset is disabled, assert the `reset` signal. Otherwise, assert the `tx_reset` and `rx_reset` signals.
2. Disable the PMA[22].
3. Trigger PMA analog reset[23].
4. Load the new Native PHY IP settings using PMA attributes directly and Avalon memory-mapped interface writes to the PCS/EMIB or use the MIF streamer.
5. Enable PMA[22]. Wait for `tx_pma_ready` and `rx_pma_ready` to assert.
6. If using traffic, enable PRBS31. Otherwise, it is optional.

---

(20) For how to enable and disable internal serial loopback, refer to *0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control*.

(21) For PMA attributes for adaptation, refer to *0x000A: Receiver Tuning Controls*.

(22) Refer to *0x0001: PMA Enable/Disable*.

(23) Refer to *PMA Analog Reset*.

7. If using a PMA configuration, load the PMA configuration using control status registers (CSR). This is loaded to the registers using PMA registers 0x200 to 0x203[(24)].
   a. Write 0x40143 = 0x80.
   b. Read 0x40144[0] until it changes to 0.
   c. Load the PMA configuration using PMA registers 0x200 to 0x203.
8. Enable internal serial loopback[(25)].
9. Run initial adaptation using PMA attributes[(26)].
10. Verify that the initial adaptation status is complete using PMA attribute code 0x0126 and data 0x0B00.
11. Enable mission mode and disable internal serial loopback (skip this step if using internal serial loopback)[(25)].
12. If not using PRBS, disable it.
13. If valid data rate traffic is available at the RX, proceed to the next step. Otherwise, rerun initial adaptation until valid traffic is available.
14. Verify that the initial adaptation status is complete using PMA attribute code 0x0126 and data 0x0B00 (skip this step if using internal serial loopback).
15. Run initial adaptation using PMA attributes[(26)].
16. Verify that the initial adaptation status is complete using PMA attribute code 0x0126 and data 0x0B00 (skip this step if using internal serial loopback).
17. If independent TX and RX reset is disabled, deassert the `reset` signal. Otherwise, deassert the `tx_reset` and `rx_reset` signals.
18. Run continuous adaptation[(26)].
19. Optional: Check the link status with `rx_is_lockedtodata` (clear the traffic checker, and verify that it is error free).
20. Start transmitting and receiving data.

Related Information

- PMA Analog Reset on page 132
- Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode on page 179
- Loading a PMA Configuration on page 204
- 0x0001: PMA Enable/Disable on page 217
- 0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control on page 220
- 0x000A: Receiver Tuning Controls on page 221
- PMA Registers 0x200 to 0x203 Usage on page 235

---

(24) Refer to *Loading a PMA Configuration* and *PMA Registers 0x200 to 0x203 Usage*.

(25) For how to enable and disable internal serial loopback, refer to *0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control*.

(26) For PMA attributes for adaptation, refer to *0x000A: Receiver Tuning Controls*.

3.1.7. Loopback modes

Loopback modes are DFT features used to verify different blocks of the transceiver PMA.

E-tile transceivers have loopback modes to debug different blocks of the transceiver. E-tile transceivers support the following loopback modes:

- Internal serial loopback
- Reverse parallel loopback

3.1.7.1. Internal Serial Loopback Path

The internal serial loopback path sets the CDR to recover data from the serializer instead of the receiver serial input pin.

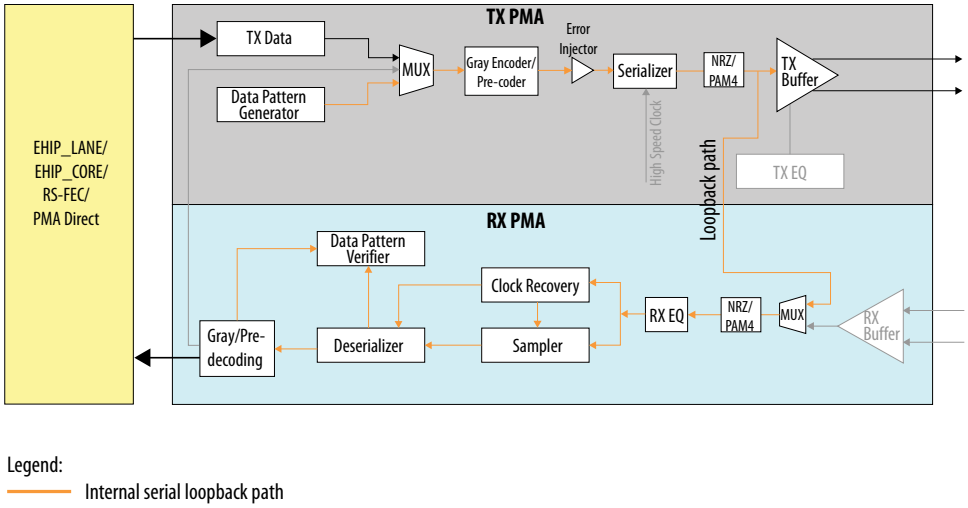
The transmitter buffer sends data normally, but internal serial loopback takes the data before the buffer.

It is implemented completely on-chip and does not require any connector on the serial path. This loopback path is enabled independently of the TX buffer enablement.

The E-tile transceiver channel also supports external loopback where you have to connect the TX differential outputs to the RX differential inputs. This external connection must consist of a transmission path with 100 Ω differential mode impedance. Mission mode is a form of external loopback where the data source is something other than the E-tile transmitter. For example, a BERT or other device's TX is providing data to the E-tile receiver.

Note: Currently, only the PRBS31/PRBS31Q patterns are supported in internal serial loopback mode.

Figure 55. Internal Serial Loopback Path



For more details on Register Read/Write support and programming, refer to *PMA Register Map* and *PMA Attribute Codes* to configure these parameters.

Related Information

- PMA Register Map on page 211
- PMA Attribute Codes on page 216

3.1.7.2. Reverse Parallel Loopback Path

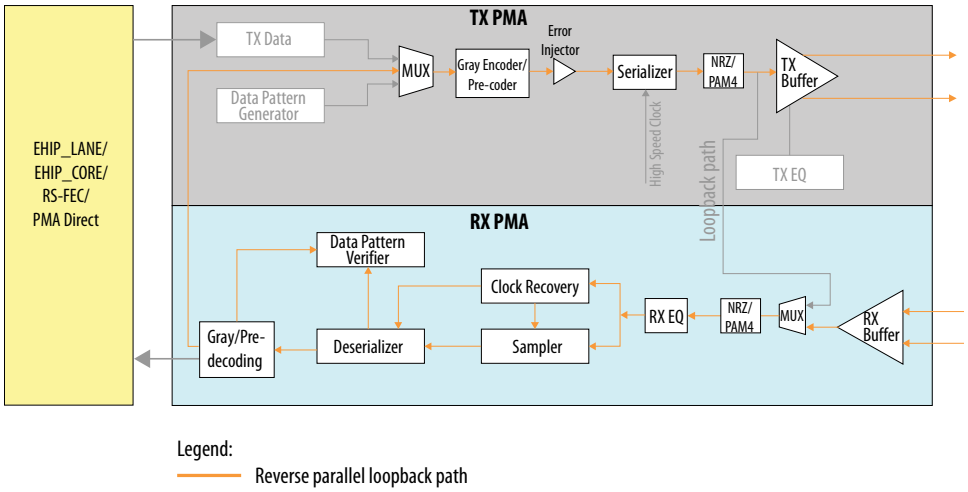
The reverse parallel loopback path sets the transmitter buffer to transmit data fed directly from the CDR recovered data.

When in reverse parallel loopback mode, the reference clock source of the received data stream must be the same reference clock that the transceiver channel receives (0ppm difference between the transmit and receive frequencies).

Additionally, the TX and RX bit rate/reference clock ratio and width mode register settings must be set to the same value to ensure proper operation of the reverse parallel loopback.

Using an external instrument, data is fed to the RX buffer, and the deserialized parallel data stream of the receiver is looped back as the parallel data input stream for the transmitter.

Figure 56. Reverse Parallel Loopback Path



Related Information

- PMA Register Map on page 211
- PMA Attribute Codes on page 216

3.1.8. PMA Interface

The PMA interface block contains the FIFO and the gearbox.

Figure 57. TX Data Flow

Simplified blocks. The FIFO is indicated in the red box.

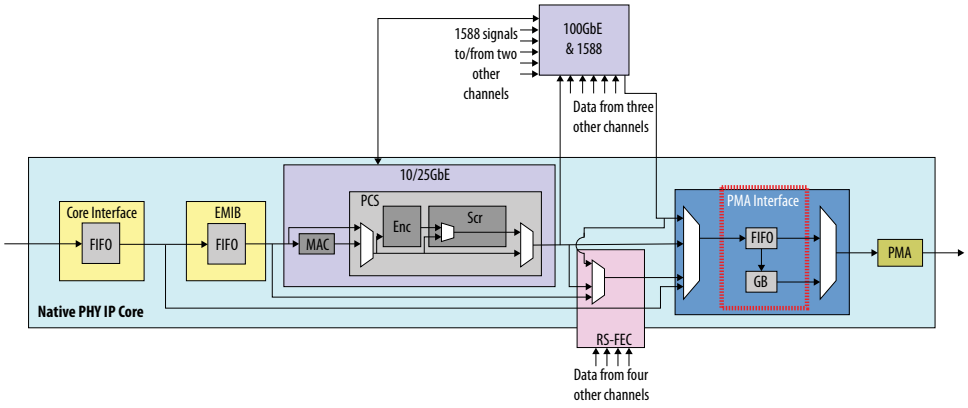
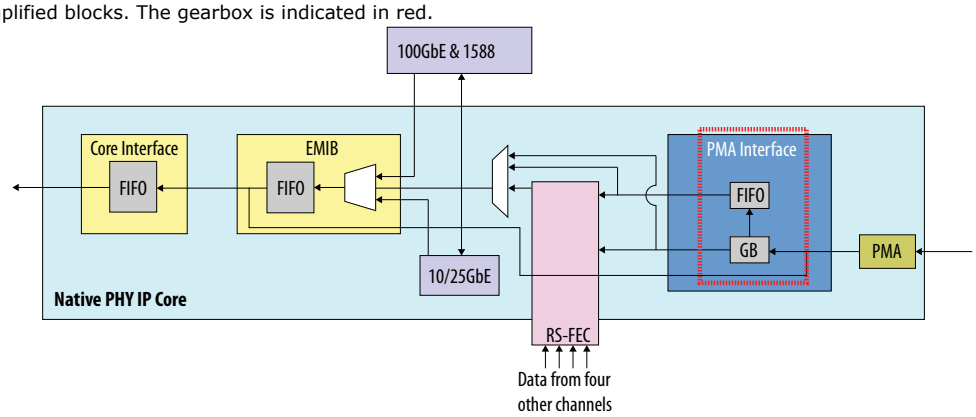


Figure 58. RX Data Flow

Simplified blocks. The gearbox is indicated in red.



The TX PMA interface FIFO is in the datapath for all modes of operation such as:

- PMA direct
- PMA direct high data rate PAM4 modes
- EHIP_LANE
- EHIP_CORE
- RS-FEC

The two TX FIFO modes are elastic or phase compensation. Elastic mode is identical to Basic mode in L/H-tiles where you can monitor the FIFO full or empty and almost full or empty signals. The control FIFO writes and reads through the read and write enable ports.

Refer to *Supported Applications/Modes* for more details about these modes, and refer to the *PMA Interface* section for more details about the E-Tile Native PHY PMA Interface.

Related Information

- Supported Applications/Modes on page 25

- PMA Interface on page 40

3.1.9. TX PMA Bonding

TX bonding enables you to minimize skew between channels.

Enable bonding by selecting the **Enable TX PMA bonding** option in the **TX PMA** tab of the Native PHY IP GUI. You can only bond channels within the same transceiver tile (24 channels). The transceiver Native PHY IP core does not support reconfiguration between bonded and non-bonded channel configurations.

The Native PHY IP core restricts the maximum number of channels to 24 for bonded configuration and provides an information message to notify you of the restriction. The fitter enforces the placement restriction to make sure all the bonded channels yield a valid placement.

TX PMA bonded channels must be placed contiguously from the bottom to the top. Use the *E-Tile Channel Placement Tool* for your channel placement.

When you enable TX PMA bonding, the maximum channel-to-channel skew is 2 UI + 125 ps.

There are two mechanisms by which to facilitate bonding:

- Transceiver interface deskew logic
- Dedicated balanced transceiver reference clock tree

Once you enable bonding in the Native PHY IP core, both of these bonding mechanisms are activated.

Related Information

E-Tile Channel Placement Tool

3.1.9.1. Transceiver Interface Deskew Logic

When you enable bonding in the Native PHY IP core for PMA Direct and high data rate PMA Direct modes only, the deskew logic in the transceiver interface aligns data transferred across multiple channels within the same clock cycle where E-tile uses the deskew bits to deskew all EMIB channels in the bonded configuration. However, the same mechanism applies even when bonding is not enabled in high data rate mode only.

For k = 0, … N-1 with N channels + single width + separate interface per channel disabled, `_data[33+k*80]` are the deskew bits.

For k = 0, … N-1 with N channels + single width + separate interface per channel enabled, `_data_ch<k>[33]` are the deskew bits.

For k = 0, … N-1 with N channels + double width + separate interface per channel disabled, `_data[33+k*80]` and `_data[73+k*80]` are the deskew bits.

For k = 0, … N-1 with N channels + double width + separate interface per channel enabled, `_data_ch<k>[33]` and `_data_ch<k>[73]` are the deskew bits.

You must perform AVMM read to the TX deskew status register, `cfg_tx_deskew_sts`, of all the bonded lanes to determine whether or not deskew is completed successfully. If it has, all the bonded channels have aligned parallel data. The deskew status register also provides further information for debugging if deskew is not successful.

`cfg_tx_deskew_sts[2] - (0x09[4]):`

- 0 = not aligned or not enabled or did't receive a deskew-bit
- 1 = aligned

`cfg_tx_deskew_sts[1:0] - (0x09[3:2]):`

- 00 = not yet received a deskew-bit
- 01 = not aligned
- 10 = received 1 set of aligned deskew-bits
- 11 = received 16 sets of aligned deskew-bits

The deskew mechanism runs continuously, so if the alignment lock is lost for some reason, monitoring `cfg_tx_deskew_sts` informs you about the status. The deskew mechanism works the same way for PMA Direct high data rate PAM4 mode for two EMIB channels. You must send the deskew pulses for the data you sent to two EMIBs and at the master transceiver interface they are aligned to before being sent to a single PMA.

3.1.9.2. Dedicated Balanced PLL Reference Clock Tree

Once bonding is enabled, use `refclk0` on the hardware. This clock is connected to the transceiver through a dedicated balanced clock tree. You do not need to do anything on Native PHY side. You can select any reference clock; however, the fitter checks that your selection on the reference clock number in the Native PHY is assigned to `refclk0` in the Intel Quartus Prime settings file (.qsf) assignments.

3.1.10. Unused Transceiver Channels

Unused transceiver channels can degrade in performance over a period of time. To preserve the performance of these unused transceiver channels, the Intel Quartus Prime software can switch the TX and RX channels on and off at a low frequency using a reference clock.

Unused channels can appear in these ways:

- Unused transceiver channels in a used tile
- Unused transceiver channels in a completely unused tile
- Unused transceiver channels in PAM4 mode
- Channels protected in order to enable reconfiguring from mission mode to channel protection mode and back again

3.1.10.1. Unused Transceiver Channels in a Used Tile

For unused channels in a transceiver tile that also includes channels that are used in the design, the Intel Quartus Prime design software selects the reference clock used by the lowest numbered channel in the design as the reference clock input for all protected channels. Therefore, you must ensure that the lowest numbered channel is

connected to a stable reference clock. This reference clock must not change dynamically. This is true for both a global assignment and instance assignment (see below). Intel Quartus Prime always gives you this critical warning:

```
Critical Warning: REFCLK refclk1ct at location PIN_AB43 is used to preserve
unused (uninstantiated) channels in the E-tile at bank 8C.
               Please ensure that this is a stable running clock and not
dynamically changing.
               If refclk1ct is a dynamically changing clock, modify your design
to connect the lowest numbered lane/channel to a stable running clock.
               See E-Tile Transceiver PHY User Guide for more information.
```

If you have met all of the reference clock requirements, you can disable this critical warning using a QSF assignment: `set_global_assignment -name MESSAGE_DISABLE <message id>`

To create the transceiver activity needed for preservation (a low-frequency, automatically generated pseudo-random bit stream) on unused channels, specify one of the following QSF assignments in the Intel Quartus Prime Settings File (`.qsf`):

- Make a global assignment (this QSF assignment preserves all unused transceiver channels in all E-tiles.): `set_global_assignment -name PRESERVE_UNUSED_XCVR_CHANNEL ON`

  Example messages seen in Intel Quartus Prime when using this assignment:

  ```
  Info: Global preservation of unused transceiver channels is enabled. All
  unused transceiver channels will be preserved.
  Info: Preserved 136 unused RX channel(s).
  Info: Preserved 136 unused TX channel(s).
  ```

- You can also do channel protection by assigning a per-pin assignment: `set_instance_assignment -name PRESERVE_UNUSED_XCVR_CHANNEL ON -to pinname`

  Example messages seen in Intel Quartus Prime when using this QSF assignment:

  ```
  Info: Channel-specific preservation of unused transceiver channels is
  enabled
  Info: Unused transceiver channel at location 'R51' will be preserved
  Info: Unused transceiver channel at location 'M48' will be preserved
  Info: Unused transceiver channel at location 'BM13' will be preserved
  Info: Unused transceiver channel at location 'BP7' will be preserved
  Info: Unused transceiver channel at location 'BD7' will be preserved
  Info: Preserved 5 unused RX channel(s).
  Info: Preserved 5 unused TX channel(s).
  ```

The `pinname` can be either the RX or the TX pin of the channel to be protected.

For example, if the `pinname` is Pin AB44, structure the per-pin assignment with the following syntax: `set_instance_assignment -name PRESERVE_UNUSED_XCVR_CHANNEL ON -to AB44`

3.1.10.2. Unused Transceiver Channels in Completely Unused Tiles

If an entire tile is completely unused, all of its transceiver channels are unused and are not present in the design netlist. They are not configured using the Native PHY IP GUI or other transceiver protocol IP GUI because they are not present in the design. These channels are present in the physical device, however, and you may want to preserve them. Preserving unused channels is necessary for all channels that you want to enable in the future.

If an entire tile is completely unused, meaning that none of the transceivers in the tile are part of the design, a reference clock must still be provided to the tile to preserve the unused transceiver channels. This means that at least one dummy channel must be configured using the Native PHY IP GUI and at least one reference clock must be provided to this transceiver channel. Then, for preserved unused transceiver channels in this otherwise unused tile, the Intel Quartus Prime design software automatically uses the reference clock input associated with the dummy channel in this tile as a reference clock to generate the pseudo-random data signal used for preservation.

In order to protect the unused transceiver channels in completely unused tiles, instantiate a single duplex channel in the Native PHY IP GUI with as low as a 2.5 Gbps data rate and its associated lowest numbered channel's reference clock. Map this dummy transceiver channel to the unused tiles in which you want to preserve channels. If you do not instantiate this dummy channel, Intel Quartus Prime gives you this message:

```
Critical Warning: User has specified PRESERVE_UNUSED_XCVR_CHANNEL global QSF
assignment, but the tile "8B" is completely empty.
You must instantiate one dummy channel in the Tile and connect to a stable
reference clock in order to preserve unused transceiver channels. See E-Tile
Transceiver PHY User Guide for more information.
```

To create the transceiver activity needed for preservation (a low-frequency, automatically generated pseudo-random bit stream) on unused channels, specify one of the following QSF assignments in the Intel Quartus Prime Settings File (`.qsf`):

- Make a global assignment (this QSF assignment preserves all unused transceiver channels in all E-tiles.): `set_global_assignment -name PRESERVE_UNUSED_XCVR_CHANNEL ON`

  Example messages seen in Intel Quartus Prime when using this assignment:

  ```
  Info: Global preservation of unused transceiver channels is enabled. All
  unused transceiver channels will be preserved.
  Info: Preserved 136 unused RX channel(s).
  Info: Preserved 136 unused TX channel(s).
  ```

- You can also do channel protection by assigning a per-pin assignment: `set_instance_assignment -name PRESERVE_UNUSED_XCVR_CHANNEL ON -to pinname`

  Example messages seen in Intel Quartus Prime when using this QSF assignment:

  ```
  Info: Channel-specific preservation of unused transceiver channels is
  enabled
  Info: Unused transceiver channel at location 'R51' will be preserved
  Info: Unused transceiver channel at location 'M48' will be preserved
  Info: Unused transceiver channel at location 'BM13' will be preserved
  Info: Unused transceiver channel at location 'BP7' will be preserved
  Info: Unused transceiver channel at location 'BD7' will be preserved
  Info: Preserved 5 unused RX channel(s).
  Info: Preserved 5 unused TX channel(s).
  ```

The `pinname` can be either the RX or the TX pin of the channel to be protected.

For example, if the `pinname` is Pin AB44, structure the per-pin assignment with the following syntax: `set_instance_assignment -name PRESERVE_UNUSED_XCVR_CHANNEL ON -to AB44`

3.1.10.3. Unused Transceiver Channels in High-Speed PAM4 Mode

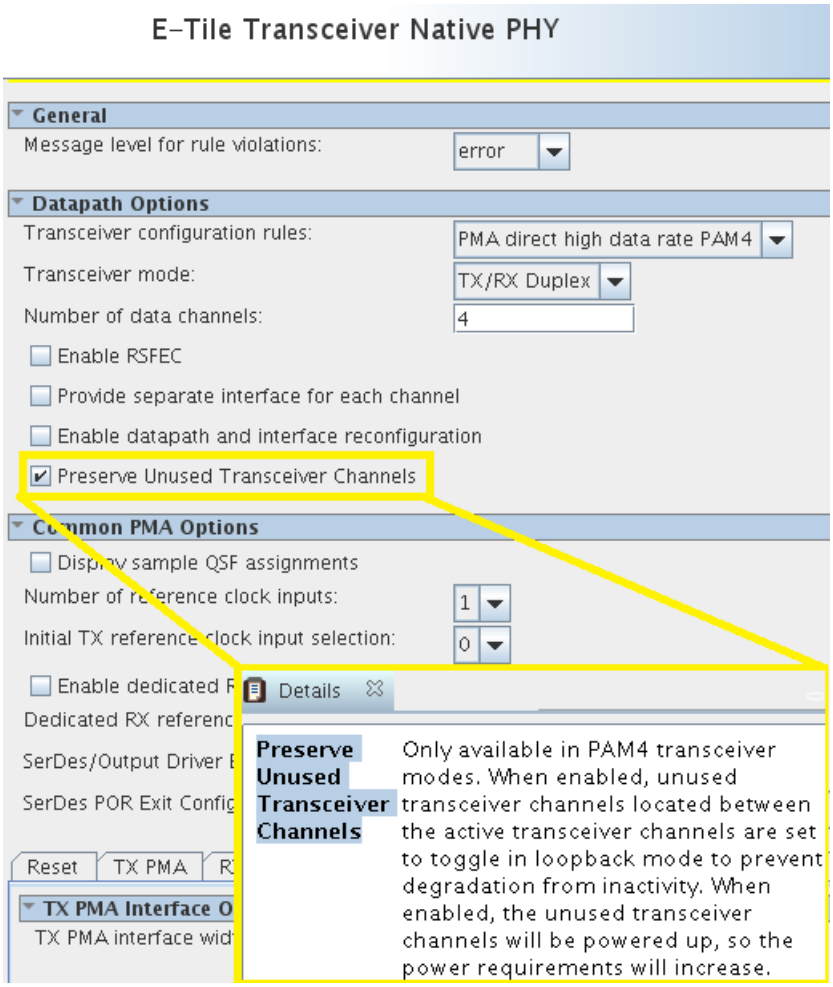
For PAM4, there is an option to preserve unused transceiver channels in the Platform Designer IP GUI. This is exclusive to the PAM4 high data rate, and the above two use cases are not applicable for PAM4.

The E-tile transceiver features a high-speed PAM4 serial transmission mode that enables effective baud rates higher than 28.9 Gbits/s. Transceiver channels that are used in high-speed PAM4 serial transmission mode must interface with the FPGA core to transmit and receive data in the parallel domain at parallel data rates that can support this high-speed serial transmission mode.

Data in the parallel domain is transferred between the FPGA core and the transceiver tile over the EMIB. This is a high-speed parallel data bus. The width of this data bus is limited by the number of physical connections between the FPGA core and the transceiver tile. Accordingly, the data transmission rate from the FPGA core to the transceiver tile is also limited by the width of the data bus and its transmission speed. For high-speed (greater than 28.9 Gbits/s) serial data transmission, a single parallel data channel of the EMIB cannot transfer parallel data fast enough to maintain the desired serial data transmission rate.

To overcome this limitation for high-speed PAM4 serial data transmission, two adjacent EMIB channels are used to transfer the parallel data between one high-speed PAM4 serial data channel and the FPGA core. The first of the two channels is referred to as the master channel and the second as the slave channel. The serial data connections for the high-speed PAM4 channel are those associated with the master channel. The slave channel serial data connections are unused in this case.

For these unused channels, a selection is provided in the Native PHY IP GUI to preserved unused transceiver channels. The GUI selection is shown below.



When the **Preserve Unused Transceiver Channels** is selected, the slave channels for the high-speed PAM4 serial transmission channels are automatically set to the desired low switching speed loopback mode. No QSF assignments are necessary for these channels. However, other than the slave channels, other channels may still need a QSF assignment to preserve the other unused channels in the tile.

This option is not available in transceiver configuration modes other than high data rate PAM4 modes.

3.1.10.4. Reconfiguring from Mission Mode to Channel Protection Mode

When you want to reconfigure an instantiated channel to channel protection mode, the reference clock configured in the design is used as the reference clock for channel protection mode. This should be a stable, running clock, and not a dynamically changing clock.

You can change these instantiated channels to channel protection mode from reconfiguration mode through a sequence of Avalon memory-mapped interface writes (see the table below) by any mechanism you have available, for example, by writing to the system console's Nios® processor.

Table 44. Steps to Configure

Step	Step Type	PMA Attribute Code	PMA Attribute Data	Transceiver Register Address	Transceiver Register Data	Comments
Disable the PMA.	PMA Attribute	0x0001	0x0000			Monitors the status and clears flags
Perform a PMA analog reset.	IP Configuration Settings			0x203:0x200	0x81000000	Monitors register 0x207
Set encoding and data width.	PMA Attribute	0x0014	0x0055			
Set the TX baud rate.	PMA Attribute	0x0005	0x1010[27]			Computes the closest valid multiplier and sends this value in the bottom eight bits of the PMA attribute data
Set the RX baud rate.	PMA Attribute	0x0006	0x1010[27]			Computes the closest valid multiplier and sends this value in the bottom eight bits of the PMA attribute data
Set PRBS control.	PMA Attribute	0x0002	0x0120			PRBS7
Set internal serial loopback.	PMA Attribute	0x0008	0x0101			
Save the values in channel addresses 0x05, 0x07, and 0x38.	Saving the mission mode settings before entering channel protection mode					
Set DCC disable.	PMA Attribute Code Read-Modify-Write			0x38[1:0]	2'b01	Disables and bypasses the DCC
Set the RX clock.	PMA Attribute Code Read-Modify-Write			0x07[7,1]	1'b0, 1'b0	Disables the RX adapter clock and RX FIFO read clock
Set the TX clock.	PMA Attribute Code Read-Modify-Write			0x05[7:2]	6'b000010	Enables the TX datapath clock
Enable the TX/RX PMA.	PMA Attribute	0x0001	0x0003			Enables TX and RX and disables the output driver

If you want to preserve the channel but not perform the steps to enter mission mode, another option is to create a preset in the Native PHY IP GUI. This preset must generate a native PHY instance with one channel or multiple contiguous channels. To properly preserve the channel, the data rate for these channels must be at least 2.5 Gbps or as close as possible to this data rate as can be achieved using an available reference clock frequency.

(27) This is a simplified set of steps. For more details, refer to *Detailed Steps for Reconfiguring from Mission Mode to Channel Protection Mode*.

Related Information

Detailed Steps for Reconfiguring from Mission Mode to Channel Protection Mode on page 292

3.1.10.5. Reconfiguring from Channel Protection Mode to Mission Mode

To reconfigure a channel from channel protection mode back to mission mode, from mission mode settings, save the values in channel addresses 0x05, 0x07, and 0x38 before entering channel protection mode.

1. Disable the PMA.
   a. Write 0x8A[7] = 0x1 to ensure that the PMA attribute status flag (0x8A[7] for the previous attribute) is cleared before writing to registers 0x84 to 0x87 to load in the new PMA attribute.
   b. Write 0x84[7:0] = 0x00.
   c. Write 0x85[7:0] = 0x00.
   d. Write 0x86[7:0] = 0x01.
   e. Write 0x87[7:0] = 0x00.
   f. Write 0x90[0] = 1'b1.
   g. Read 0x8A[7]. It should be 1.
   h. Read 0x8B[0] until it changes to 0.
   i. Write to 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
   j. Wait for `tx_pma_ready` or `rx_pma_ready` to deassert.
2. Reset the internal controller inside the PMA.
   a. Write 0x200[7:0] = 0x00.
   b. Write 0x201[7:0] = 0x00.
   c. Write 0x201[7:0] = 0x00.
   d. Write 0x203[7:0] = 0x81.
   e. Read 0x207 until it reads 0x80.
3. Change the TX and RX channel PMA attributes back to the state they were in in mission mode.
   a. Set attribute 0x0002 to 0x03FF to disable internal PRBS.
      i. Write 0x84[7:0] = 0xFF.
      ii. Write 0x85[7:0] = 0x03.
      iii. Write 0x86[7:0] = 0x02.
      iv. Write 0x87[7:0] = 0x00.
      v. Write 0x90[0] = 1'b1.
      vi. Read 0x8A[7]. It should be 1.
      vii. Read 0x8B[0] until it changes to 0.
      viii. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
   b. Set attribute 0x0008 to 0x0100 to enter the external loopback mode.

i. Write 0x84[7:0] = 0x00.
ii. Write 0x85[7:0] = 0x01.
iii. Write 0x86[7:0] = 0x08.
iv. Write 0x87[7:0] = 0x00.
v. Write 0x90[0] = 1'b1.
vi. Read 0x8A[7]. It should be 1.
vii. Read 0x8B[0] until it changes to 0.
viii. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.

c. Write 0x91[0] = 1'b0 to restore the other attributes to their mission mode values.
d. Restore register values of registers 0x38, 0x05, and 0x07 back to the state they were in at mission mode.

4. Restore the channels to re-enable PMA.
   a. Write 0x84[7:0] = 0x07.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x01.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
   i. Wait for `tx_pma_ready` or `rx_pma_ready` to assert.
5. Perform initial adaptation for the restored channels.

3.1.11. Low Power Mode (LPM)

The E-tile device powers up in normal power mode. Thus, to use low power mode, configure and calibrate the E-tile device and enter mission mode before manually enabling low power mode.

Enabling and Disabling Low Power Mode

Use registers 0x200-0x203 to enable LPM. LPM is enabled one channel at a time. Use the base address for the desired channel when doing the register writes to enable LPM for that calling channel. For example, to load to channel 0:

1. Write 0x01 to 0x200 to enable LPM, or write 0x00 to 0x200 to disable LPM.
2. Write 0x00 to 0x201 as this should be all 0's.
3. Write 0x00 to 0x202 to target the calling channel (channel 0 in this case).
4. Write 0x98 to 0x203 to use the OPCODE for LOW_POWER_MODE. See the "Loading PMA Configuration Register LOW_POWER_MODE" figure for details.

While in Low Power Mode

- LPM determines the mode of the implementation and turns off unused IP and DFT.
- Initial adaptation and continuous adaptation function normally.
- Adaptive settings perform normally.
- Mission mode performance is unaffected.
- There are no timing constraints or requirements.
- There are no issues with changing PMA static settings.

When to Enable or Disable Low Power Mode

- Disable LPM before changing the functional mode (from NRZ to PAM4 or from PAM4 to NRZ). Re-enable LPM after the change.
- Enabling and disabling LPM does not impact performance.
- Disable LPM before disabling the transceiver. If you do not, you must perform a PMA analog reset to recover the transceiver.
- A PMA analog reset exits LPM, so re-enable LPM after performing adaptation that requires a PMA analog reset.
- Intel recommends disabling LPM for DFT functions such as internal serial loopback, reverse parallel loopback, and eye measurements.
- However, when DFT is utilized in LPM, the associated IP is turned on. To regain the best power performance, re-enable LPM.

How to Use Internal Serial Loopback

Internal serial loopback mode does not work in conjunction with LPM because the phase interpolator is disabled and there is no activity in the receiver. To use internal serial loopback mode:

1. Disable LPM.
2. Enable internal serial loopback mode.
3. Perform testing.
4. Disable internal serial loopback mode to enter mission mode.
5. Re-enable LPM.

Unused Transceiver Channel Implementation with Low Power Mode

1. Configure unused transceiver channels in internal serial or reverse parallel loopback mode. See *Unused Transceiver Channels* for details.
2. Enter LPM. See "Enabling and Disabling Low Power Mode" above and the "Loading PMA Configuration Register LOW_POWER_MODE" figure for details.
3. Enable PLL (this can be part of the LPM register writes). See the "Loading PMA Configuration Register LOW_POWER_MODE" figure for details. Or use PMA attribute 0x8092 0x188C.

RX and TX toggle as normal, but the data may have errors.

Related Information

- Unused Transceiver Channels on page 95
- Enabling Low Power Mode for Multiple Channels on page 205

- PMA Registers 0x200 to 0x203 Usage on page 235
  See the "Loading PMA Configuration Register LOW_POWER_MODE" figure.

3.2. Physical Coding Sublayer (PCS) Architecture

The E-tile PCS is located in the EHIP_LANE block, which includes the following features:

- 64B/66B encoder/decoder
- Scrambler/descrambler
- Block distribution/block synchronization
- Lane reorder

The PCS features are not available within the Native PHY IP core. Refer to the *E-Tile Hard IP for Ethernet Intel FPGA IP User Guide* for details about the EHIP_LANE block.

Related Information

E-Tile Hard IP for Ethernet Intel FPGA IP User Guide

3.3. Reed Solomon Forward Error Correction (RS-FEC) Architecture

The E-tile includes a Reed Solomon Forward Error Correction (RS-FEC) block.

For more basic RS-FEC information, refer to *AN 846: Intel Stratix 10 Forward Error Correction*.

The RS-FEC core supports the following standards:

- 100GbE: IEEE 802.3 Clause 91
- 100GbE with KP-FEC: IEEE 802.3 Clause 91
- 128GFC: Fibre Channel Framing and Signaling - 4 (FC-FCS-4) Clause 5.6
- 25GbE: IEEE 802.3 Clause 108
- 32GFC: Fibre Channel Framing and Signaling - 4 (FC-FCS-4) Clause 5.4

100GbE with KP-FEC uses two physical PAM4 coded lanes, also called, 100 Gigabit Attachment Unit Interface (CAUI-2). It uses the RS(544,514) FEC. The two physical lanes are supported by bit-multiplexing the RS-FEC core's four PMA lanes pairwise outside of the RS-FEC core. The remaining defined clients use the RS(528,514) FEC.

In the CPRI standard, the CPRI FEC refers to 32GFC. CPRI is like 32GFC except for the line rate, which is 24 Gbps.

Table 45. Supported FEC Specifications in E-Tiles

Supported RS-FEC Type	Compliance
RS-FEC (528, 514) RS-FEC (544, 514)	IEEE 802.3 Clause 91

Table 46. FEC Details in E-Tiles

Resource	Description
Number of RS-FEC blocks per E-tile	6
Number of RS-FEC lanes per FEC block	4
RS-FEC block implementation	Hard
RS-FEC block locations	Between the transceiver interface and Ethernet Hard IP (EHIP_TOP)

Related Information

AN 846: Intel Stratix 10 Forward Error Correction

3.3.1. RS-FEC Modes

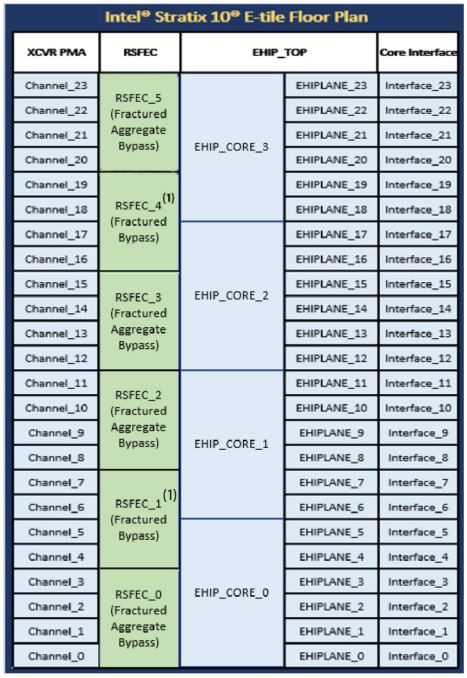
Table 47. Example Applications for Various FEC Modes

Supported RS-FEC Modes	RS-FEC Receives Data From	Example Applications	Details
Fractured	EHIP_LANE	25GbE - NRZ w/FEC (528, 514)	You can configure all six FEC blocks per E-tile in this mode. NRZ mode: Four lanes within a FEC block operate independently for single lane protocols. However, RS-FEC needs to be shared among the NRZ channels within one IP instantiation.
	FPGA core	CPRI 24G - NRZ w/ FEC (528, 514) 32GFC w/ FEC (528, 514)	
Aggregate	EHIP_CORE	100GbE (4 x 25G) – NRZ w/ FEC (528, 514) 100GbE (2 x 50G) – PAM4 w/ FEC (544, 514)	You can configure a maximum of four out of six FEC blocks per E-tile in this mode. Refer to Figure 59 on page 106 for more details. Four lanes of a FEC block are used together for multi-lane protocols, like 100GbE.
	PMA Direct	128 GFC	
Bypass	—	10GbE – NRZ 25GbE - NRZ w/o FEC 100GbE (4 x 25G) – NRZ w/o FEC	Protocols or applications that do not need RS-FEC

You can configure RS-FEC blocks in many possible combinations depending on your application requirements.

Figure 59. E-Tile Floor Plan Configurations

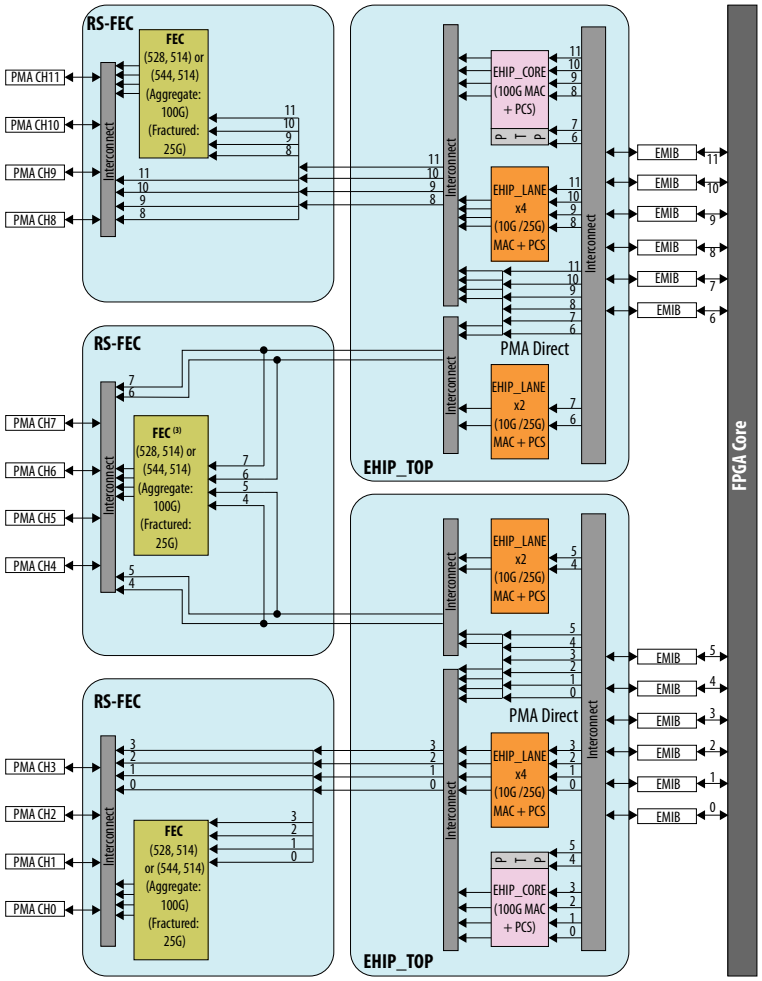
This figure illustrates the placement of various architecture blocks, and the modes supported in the RS-FEC blocks.



Note:
1. This block cannot be used in combination with EHIP_CORE - fractured bypass.

Figure 60. Datapath Routing for RS-FEC Configurations

Supported RS-FEC configurations are shown inTable 47 on page 105.



Legend:

= EHIP_CORE = FEC = EHIP_LANE

Notes:

1. Not all datapath combinations are available.
2. Datapath enablement depends on the configuration you are implementing. Refer to the E-Tile Channel Placement tool for possible configurations.
3. This FEC block can only be used in aggregate mode with FEC direct application (e.g. 128GFC Fibre-Channel). This FEC block cannot be used in in aggregate mode with EHIP_CORE because there is no EHIP_CORE in this location.

Table 48. FEC Block Modes by Channel

This table corresponds to Figure 61 on page 108 and Figure 62 on page 109

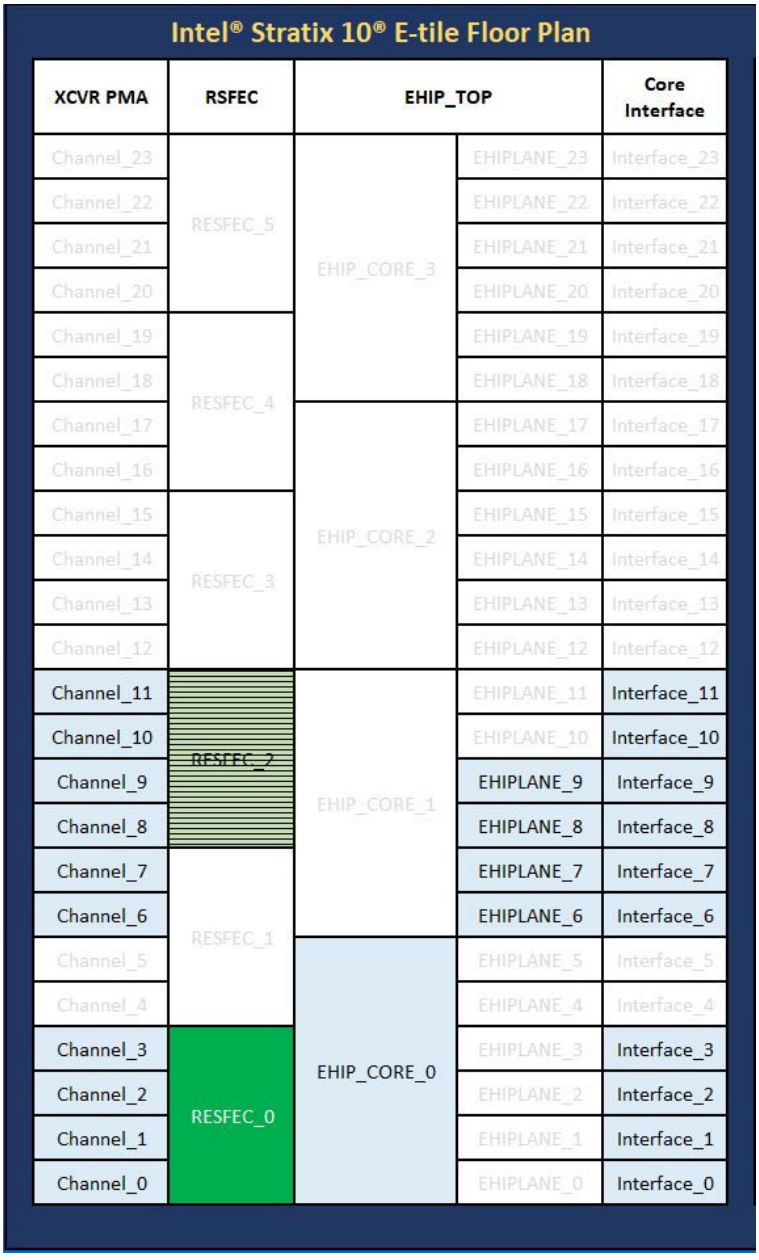
Channels	FEC Block Mode	FEC Receives Data From
0 to 3	Aggregate	EHIP_CORE
4 to 5	Bypass	N/A

continued...

Channels	FEC Block Mode	FEC Receives Data From
6 to 7	Bypass	EHIP_LANE
8 to 9	Fractured	EHIP_LANE
10 to 11	Fractured	FPGA core

Figure 61. Example Channel Configurations Implementing Various FEC Modes using the E-Tile Channel Placement Tool

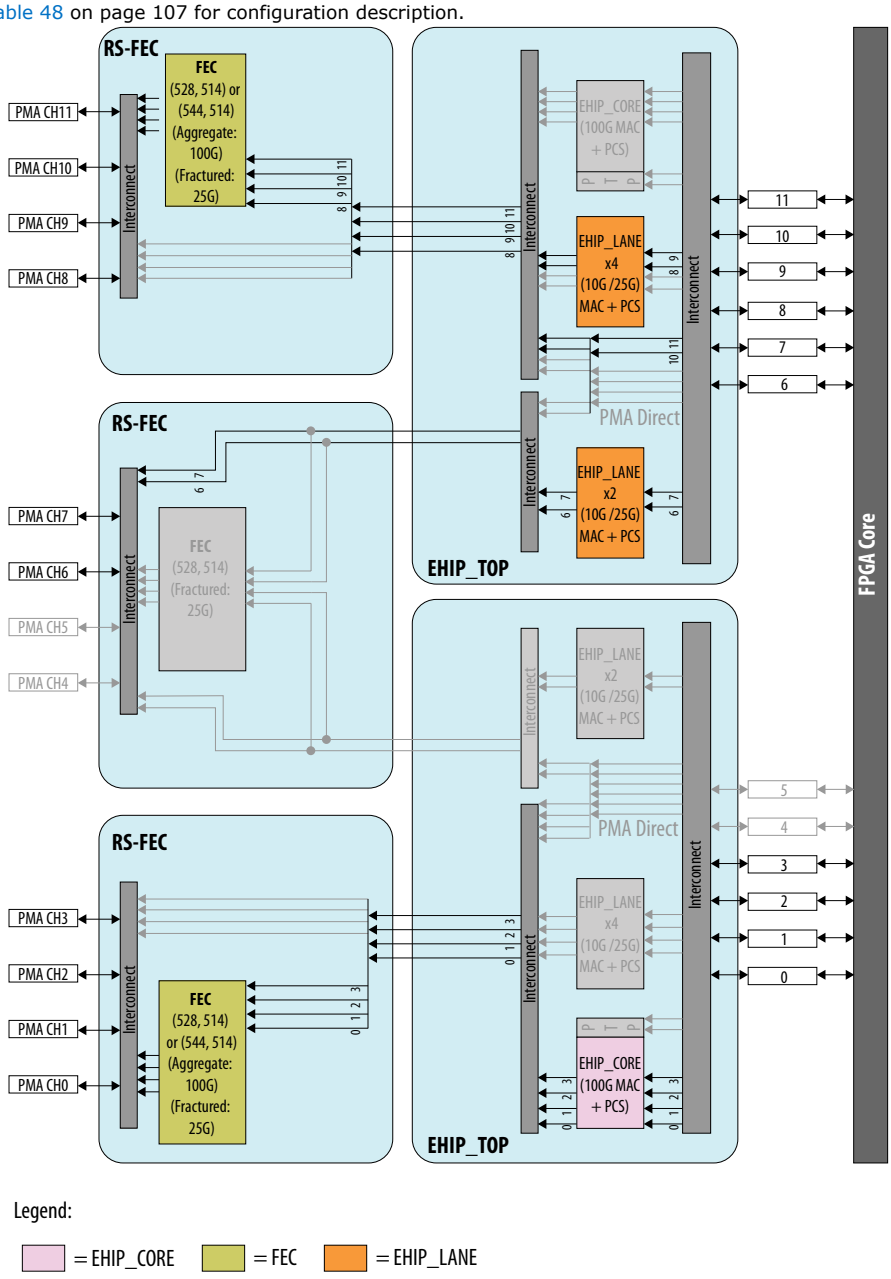
Your implementation may vary depending on your intended application. Check Table 48 on page 107 for configuration description.



Refer to the *E-Tile Channel Placement Tool* for details about possible channel placement based on system requirements.

Figure 62. Channel Configurations Implementing Various FEC Modes

Check Table 48 on page 107 for configuration description.



Related Information

E-Tile Channel Placement Tool

3.4. E-Tile Transceiver PHY Architecture Revision History

Document Version	Changes
2020.01.31	Made the following changes: • Updated the "PMA Parameter Tuning for Extending Dynamic Range" table. • Added *Duplex Adaptation Flow*. • Added *RX Simplex Adaptation Flow*. • Added *Dynamic Reconfiguration Adaptation Flow*. • Updated instructions for preserving *Unused Transceiver Channels in a Used Tile*. • Updated instructions for preserving *Unused Transceiver Channels in Completely Unused Tiles*. • Updated instructions for preserving *Unused Transceiver Channels in High-Speed PAM4 Mode*. • Added *Reconfiguring from Mission Mode to Channel Protection Mode*. • Added *Reconfiguring from Channel Protection Mode to Mission Mode*.
2019.10.11	Made the following changes: • Updated *Low Power Mode*. • Added the "Step Size" column to the "TX Equalization Settings for PAM4 and NRZ Signals" table. • Added the *E-Tile Transmitter Equalization Tool*. • Clarified "PMA Parameter Tuning for Extending Dynamic Range." • Added the Related Information links for the Intel Agilex device documents.
2019.07.29	Made the following changes: • Added *Low Power Mode*. • Changed "internal or serial loopback" to "internal serial loopback." • Added "PMA Initial Adaptation Effort Status" descriptions. • Added a description of the mission mode to *Internal Serial Loopback Path*. • Limited the sum of the TX equalization settings to less than or equal to 26. • Limited the POST equalization setting to even values. • Removed the limitation on the PRE2 equalization setting to even values. • Added a description of the equalization tables: These tables are intended to look up a given combination of transmitter equalization settings. The Input columns are the input values you enter, and the Outputs columns are the actual output values after programming. You can look up whether your input-output combination is valid or invalid and whether your inputs match your outputs.
2019.04.19	Made the following changes: • Added this note to *Programmable Termination Modes*: If the above requirements are not met, a typical value of 100 nF of capacitive termination can be used on the board. • Inserted under Fractured in the "Example Applications for Various FEC Modes" table: However, RS-FEC needs to be shared among the NRZ channels within one IP instantiation. • Added *Unused Transceiver Channels in Completely Unused Tiles*. • Added *Unused Transceiver Channels in High-Speed PAM4 Mode*.
2019.03.13	Made the following change: • Added *TX PMA Bonding* instruction: "TX PMA Bonded channels must be placed contiguously from the bottom to the top. Use the *E-Tile Channel Placement Tool* for your channel placement."
2019.03.07	Made the following change: • Changed the data rate for E-tile Non-Return to Zero (NRZ) to 28.9 Gbps.
2019.02.04	Made the following changes: • Added *PMA Reset*. • Updated the "PMA Parameter Description and Range" table. • Updated the "PMA Parameter Tuning for Extending Dynamic Range" table.
2018.10.08	Made the following changes: • Changed the rule for the postcursor in the "TX Equalization Settings for NRZ Signals" table. • Re-titled the "Receiver Equalizer" section to "RX Adaptation Modes." • Added the "PMA Tuning" section and all subsections. • Added the "External Loopback Path" section.

continued...

Document Version	Changes
	• Added supported standards by RS-FEC in the "Reed Solomon Forward Error Correction (RS-FEC) Architecture" section. • Added further description about the absolute maximum transceiver input in the "Programmable Termination Modes" section. • Changed the description for the initial adaptation mode in the "E-Tile Receiver PMA RX Adaptation Modes" table. • Added a note about the gearbox in the "PMA Interface" section. • Clarified that PCS features are not available within the Native PHY IP core in the "Physical Coding Sublayer (PCS) Architecture" section. • Added PMA Direct to the aggregate mode in the "Example Applications for Various FEC Modes" table.
2018.08.08	Made the following changes: • Changed the equation for TX equalization for NRZ signals in the "TX Equalizer" section.
2018.07.18	Made the following changes: • Added the "E-Tile Receiver PMA RX Adaptation Modes" table. • Updated the "Data Pattern Generation" section and added the "QPRBS13-CEI Pattern" figure. • Updated the description of the PMA interface in the "PMA Interface" section.
2018.05.15	Made the following changes: • Added the "PCS" section. • Updated the "PMA Architecture Block Diagram" figure. • Updated the "Internal Serial Loopback Path" figure. • Updated the "Reverse Parallel Loopback Path" figure. • Removed the **Termination modes** parameter from the "PMA Transmitter Programmable Parameters" table. • Added the following sections: — "TX PMA Bonding" — "Transceiver Interface Deskew Logic" — "Dedicated Balanced PLL Refclk Tree" • Added links to the NRZ—Transmitter Equalization Tool and PAM4—Transmitter Equalization Tool in the "TX Equalizer" section.
2018.01.31	Initial release.

4. Clock Network

The E-tile transceivers are equipped with the following clock networks:

- Reference clock
- Core interface clock

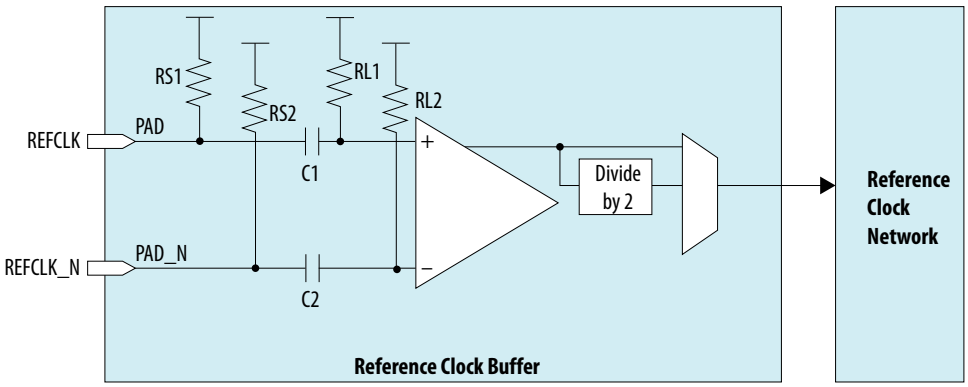
4.1. Reference Clock Pins

There are a maximum of nine LVPECL reference clock pins on every E-tile. Refer to the respective *Pin-Out Files for Intel FPGA Devices* to find the actual number of reference clocks available in each device. You can configure the pins as either 2.5-V LVPECL compliant or 3.3-V LVPECL tolerant. You can select between 2.5 V and 3.3 V using the QSF statements defined in *QSF Assignments for Reference Clock Pins*. There are source terminations (RS1 and RS2, 50 Ω each) and load terminations (RL1 and RL2, 250 Ω each), as well as built-in internal AC coupling for differential reference clock input pairs. AC coupling is always enabled irrespective of internal or external termination. Intel recommends using the default setting, which is internal source termination at 2.5 V. Carefully disable internal source termination only when you need external termination at 3.3 V (or 2.5 V). Source termination RS1/2 are bypassed when enabling external termination. For external termination and related reference clock detailed requirements, refer to the *Device Family Pin Connection Guidelines*. Refer to *Device Data Sheet* for more details on supported LVPECL specifications and required reference clock specifications.

The E-tile transceiver reference clock input pin supports a frequency range of 125 MHz to 700 MHz, but the reference clock network supports a maximum frequency of 500 MHz. Whenever you configure a reference clock frequency of greater than 500 MHz, the **Divide by 2** block is automatically instantiated along with the IP instantiation.

The hardware supports nine reference clocks pins, but the Native PHY IP core parameter editor provides any five reference clocks for a given design implementation. You select which five based on your board layout.

Figure 63. IO Pad Ring - Transceiver Reference Clock Input Pad



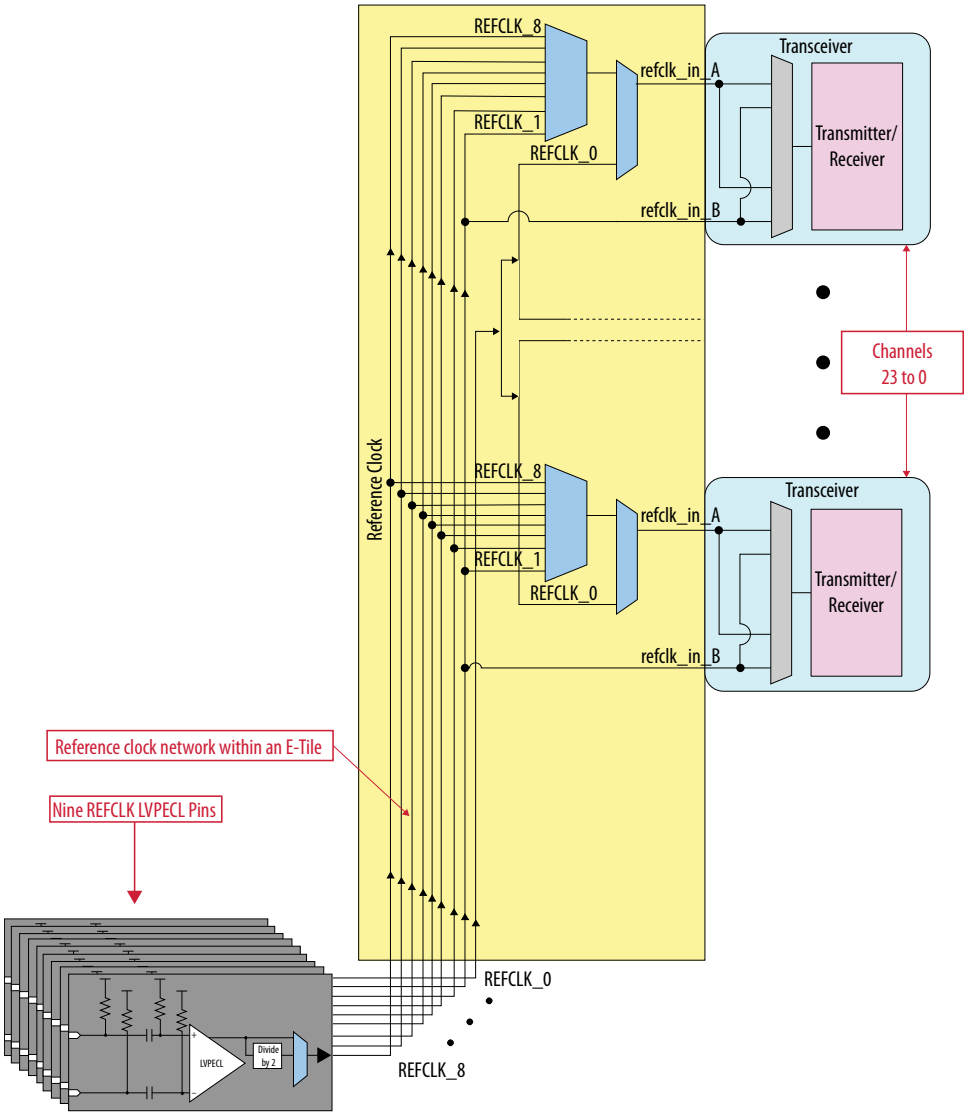
E-tile completes the power-up configuration successfully provided that a valid reference clock frequency, 125 MHz - 500 MHz (if the `refclk` **Divide by 2** is disabled) or 250 MHz - 700 MHz (if the `refclk` **Divide by 2** is enabled), is available during device power-up, which may or may not be the same as what is configured in the transceiver IP. A difference in the configured `refclk` in the IP compared to the available `refclk` on the board can cause unexpected transitions on the E-tile TX output.

Make sure you are okay with this behavior until the `refclk` frequencies are set correctly followed by the recommended reset and device configuration steps as per *PMA Analog Reset*. If the unexpected transitions are not acceptable, you can disable the transceiver TX output by writing the attribute code 0x0001 with data 0x0003 after power-up. The E-tile TX may still give some unexpected transitions between the power-up phase until the attribute code 0x0001 is written.

After correctly configuring back the on-board reference clock, follow the recommended reset and device configuration steps as per *PMA Analog Reset* to reset the internal controller. Refer to the *Register Map* for more details on attribute codes and data. Not having a stable reference clock during device configuration causes the configuration to fail.

Figure 64. REFCLK LVPECL Pins

This diagram illustrates the nine `refclk` pins and the reference clock network within a given E-tile.



For details on LVPECL standard spec, refer to *Device Data Sheet*.

Related Information

- QSF Assignments for Reference Clock Pins on page 115
- PMA Analog Reset on page 132
- Register Map on page 211
- Pin-Out Files for Intel FPGA Devices
- Intel Stratix 10 Device Data Sheet
- Intel Stratix 10 Device Family Pin Connection Guidelines
- Intel Agilex Device Data Sheet

- Intel Agilex Device Family Pin Connection Guidelines

4.1.1. QSF Assignments for Reference Clock Pins

Refer to the *Device Family Pin Connection Guidelines* for how to connect unused reference clock pins.

Table 49. QSF Assignments for a Single Reference Clock Pin (refclk[0])

You must manually include these QSF settings for every used reference clock pin.

Description	Value	QSF Assignment
Set reference clock IO standard	differential LVPECL	`set_instance_assignment -name IO_STANDARD "DIFFERENTIAL LVPECL" -to <refclk_name> -entity <block_name>`
Enable on-die termination resistors	`enable_term` = (on-chip termination on) `disable_term` = (on-chip termination off)	`set_instance_assignment -name HSSI_PARAMETER "refclk_divider_enable_termination=enable_term" -to ref_clk[0]` Recommendation: Set this to `enable_term` unless external on-board termination is used and internal termination is supposed to be bypassed.
Select 3.3 V tolerant instead of 2.5 V	`enable_3p3v_tol` = (3.3V) `disable_3p3v_tol` = (2.5V)	`set_instance_assignment -name HSSI_PARAMETER "refclk_divider_enable_3p3v=enable_3p3v_tol" -to ref_clk[0]`[28] Recommendation: Set this to `disable_3p3v_tol` unless the clock source is compliant to LVPECL 3.3v standard.
Enable LVPECL driver hysteresis	`enable_hyst` = (hysteresis on) `disable_hyst` = (hysteresis off)	`set_instance_assignment -name HSSI_PARAMETER "refclk_divider_disable_hysteresis=enable_hyst" -to ref_clk[0]` Recommendation: Set this to `disable_hyst` provided that the reference clock characteristic meets the specification in the *Device Data Sheet*.
Set reference clock frequency	`freq_in_Hz` = "legal value"	`set_instance_assignment -name HSSI_PARAMETER "refclk_divider_input_freq=freq_in_Hz" -to ref_clk[0]` Recommendation: Use the same reference clock frequency number as in the respective transceiver IP.
Power down LVPECL driver	`false` = (driver on) `true` = (driver off)	`set_instance_assignment -name HSSI_PARAMETER "refclk_divider_powerdown_mode=false" -to ref_clk[0]` Recommendation: If you plan to use the target reference clock, set this to `false`.

Related Information

- Intel Stratix 10 Device Data Sheet
- Intel Stratix 10 Device Family Pin Connection Guidelines
- Intel Agilex Device Data Sheet
- Intel Agilex Device Family Pin Connection Guidelines

(28) Refer to the *Device Data Sheet* for the reference clock voltage rating electrical specifications.

(29) Refer to the *Device Data Sheet* for the reference clock frequency specification.

4.1.2. Dynamic Reconfiguration of Reference Clock

Reference clocks can be dynamically reconfigured in two ways:

- Change the reference clock frequency on a single reference clock pin.
- Change the reference clock frequency by switching between reference clock pins.

Refer to *Dynamic Reconfiguration Flow for Special Cases* for reference clock dynamic reconfiguration steps.

Related Information

Dynamic Reconfiguration Flow for Special Cases on page 161

4.2. Core Clock Network Use Case

These use cases provide guidance about how you can connect various clocks through the GUI for different use cases.

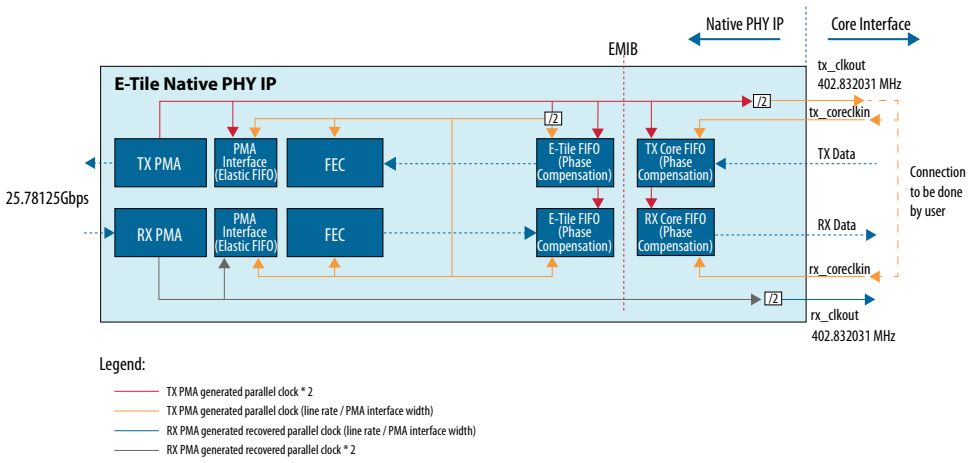
4.2.1. Single 25 Gbps PMA Direct Channel (with FEC) Within a Single FEC Block

Table 50. Single 25 Gbps PMA Direct Channel (with FEC) Within a Single FEC Block Configuration

Data Rate	TX and RX Double Width	PMA Interface	Core Interface	tx_clkout Clock Source
25.78125 Gbps	Enabled	32 bits	64 bits	Half-Rate

For FIFO in Phase Compensation mode, connect half rate `tx_clkout` (402.832031 MHz, that is, 25.78125 Gbps/64) to `tx_coreclkin` and `rx_coreclkin`. If you use any other source for `tx_coreclkin`, make sure `tx_coreclkin` has 0 PPM difference with `tx_clkout`.

Figure 65. PMA Direct with FEC



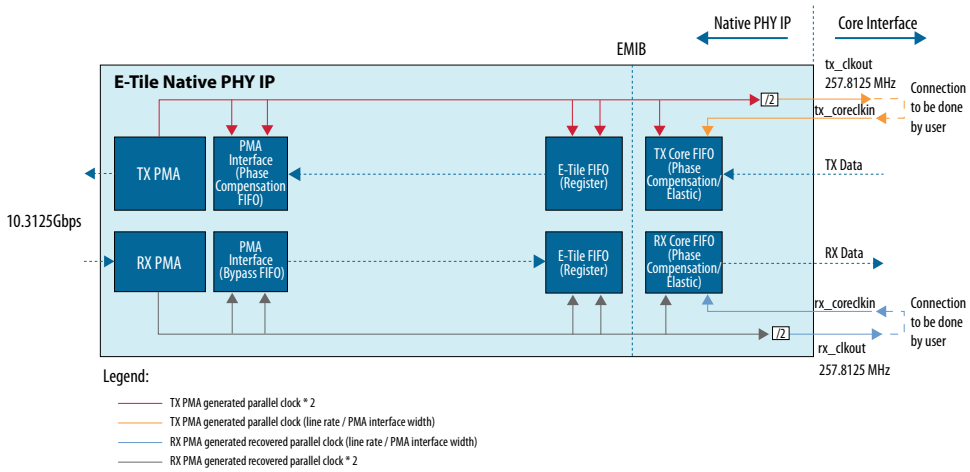
4.2.2. Single 10 Gbps PMA Direct Channel (without FEC)

Table 51. Single 10 Gbps PMA Direct Channel Configuration

Data Rate	TX and RX Double Width	PMA Interface	Core Interface	tx_clkout Clock Source
10.3125 Gbps	Enabled	20 bits	40 bits	Half-Rate

For Core Interface FIFO in Phase Compensation mode, connect half rate `tx_clkout` (257.8125 MHz) to `tx_coreclkin` and connect `rx_clkout` (257.8125 MHz) to `rx_coreclkin`. If you use any other source for `tx_coreclkin/rx_coreclkin`, make sure `tx_coreclkin` and `rx_coreclkin` have 0 PPM difference with `tx_clkout` and `rx_clkout`, respectively. E-Tile FIFOs in this mode are used in REGISTER mode.

Figure 66. PMA Direct 10G x 1



4.2.3. Four 25 Gbps PMA Direct Channel (with FEC) within a Single FEC Block

Table 52. Four 25 Gbps PMA Direct Channel (with FEC) within a Single FEC Block Configuration

Data Rate per Channel	Number of Channels	TX and RX Double Width	PMA Interface	Core Interface	tx_clkout Clock Source
25.78125 Gbps	4	Enabled	32 bits	64 bits	Half-Rate

Connect half rate `tx_clkout` (402.832031 MHz) to the `tx_coreclkin` and `rx_coreclkin`. If you use any other source for `tx_coreclkin`, make sure `tx_coreclkin` has 0 PPM difference with `tx_clkout`.

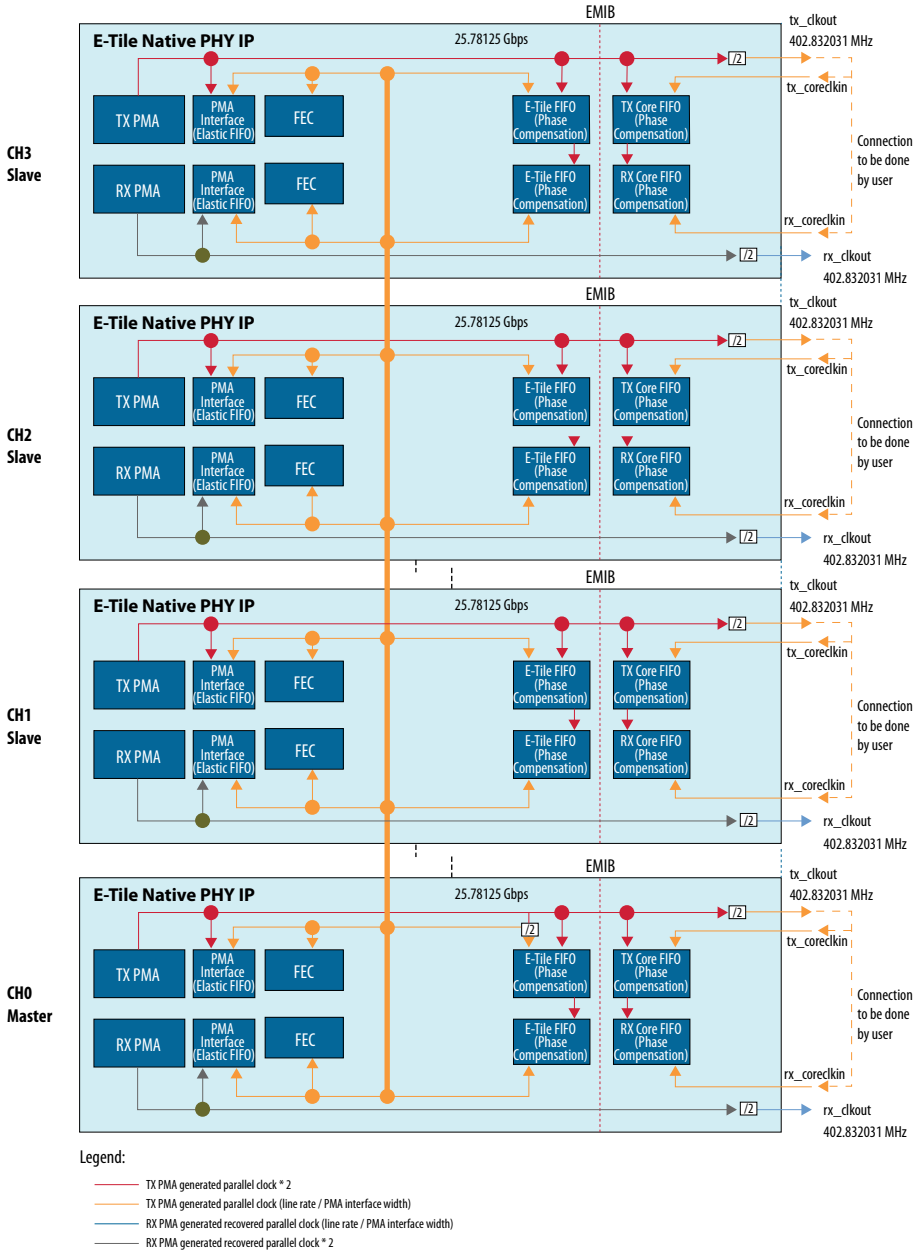
4.2.3.1. Master-Slave Configuration: Option 1

All four channels use a common FEC block, but FEC only uses one clock from the four available channels. You can select the source channel of the FEC clock in the **FEC** tab of Native PHY IP Parameter Editor through the **RS-FEC Clocking Mode** option. The selected source channel is considered the master. The other three channels use that

same clock for clocking their TX and RX data paths, and are considered slave channels. An interruption on the master channel PMA, a PMA reset, for example, impacts the slave channels. This creates a dependency between the master and the slave channels. The figure below assumes that all four channels have a common reference clock source (0 PPM between all four channels).

Figure 67. PMA Direct 25 Gbps x 4 (FEC On)

RS-FEC is also clocked by the TX PMA generated clock.



4.2.3.2. Master-Slave Configuration: Option 2

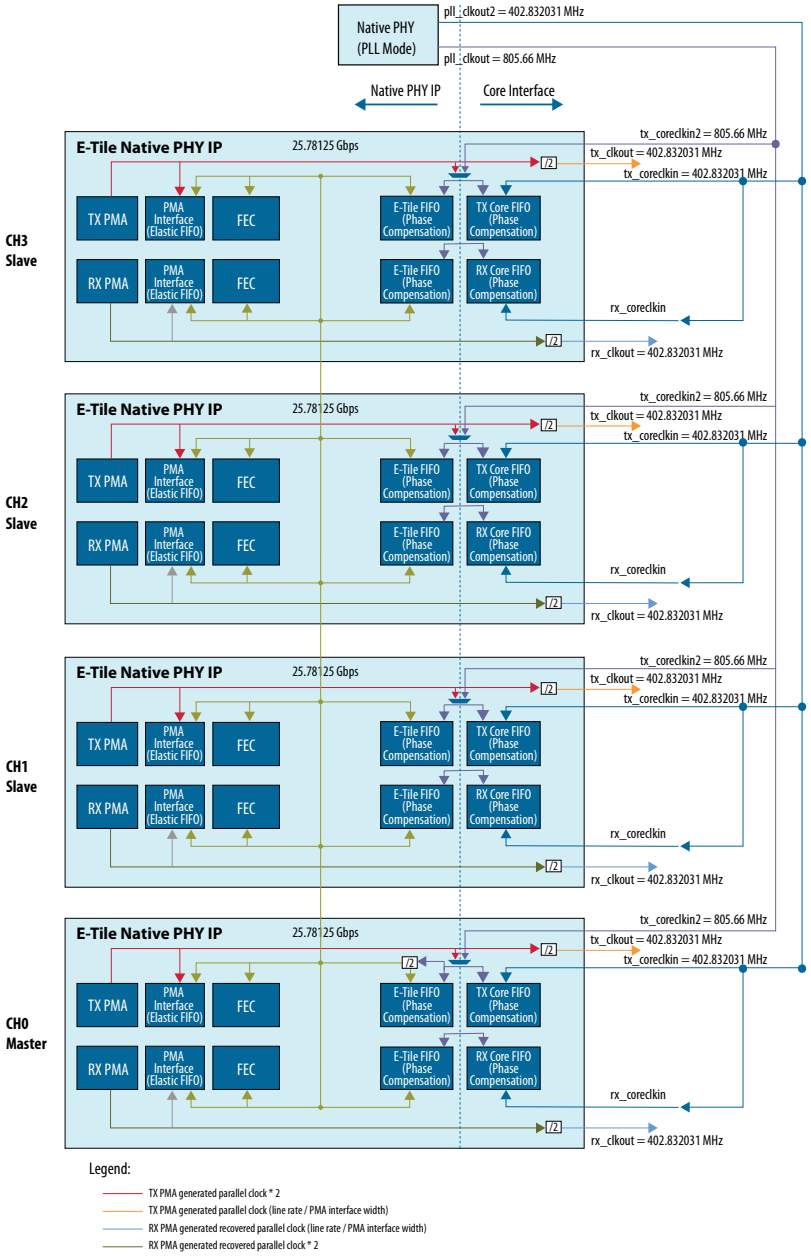
This configuration is also referred as external EMIB clocking. In this configuration, you can select to import the TX and RX datapath clocks and EMIB clock from a separate transceiver channel. Enable this by selecting the `tx_coreclkin2` port and by selecting **Enable external clock mode** in the **Core Interface** tab of Native PHY IP Parameter Editor. Once `tx_coreclkin2` is enabled, an extra input port is exposed in the core to drive the individual EMIB clock for each 25 Gbps channel. The FEC clock is still provided by the Master channel. This method removes the dependency of a PMA reset between the master and slave channels. The E-tile transceivers can be used in PLL mode to supply a clock in this configuration as shown below. Only transceivers in PLL mode can be used for clocking all four 25G channels. When a transceiver is in PLL mode, you cannot use it for TX or RX operations. Using external EMIB clocking also helps provide a clock to low data rate channels when different data rate channels are placed in same FEC block, for example, 25GE and 24G CPRI. The following figure shows one master 25 Gbps channel providing the datapath clock to other three slave 25 Gbps channels.

In this external EMIB clocking configuration, you must read the `pll_locked` output from the PLL channel before resetting the transceiver channel:

- Wait until `pll_locked` from the PLL channel is asserted before deasserting the transceiver channel reset at power-up.
- If `pll_locked` from the PLL channel is deasserted at any time, hold the respective transceiver channels in reset until `pll_locked` is reasserted.

The reference clock for the PLL channel should have the same reference clock source as the reference clock for the Ethernet channel to have 0 PPM.

Figure 68. PMA Direct 25 Gbps x 4 (FEC On) Independent Configuration



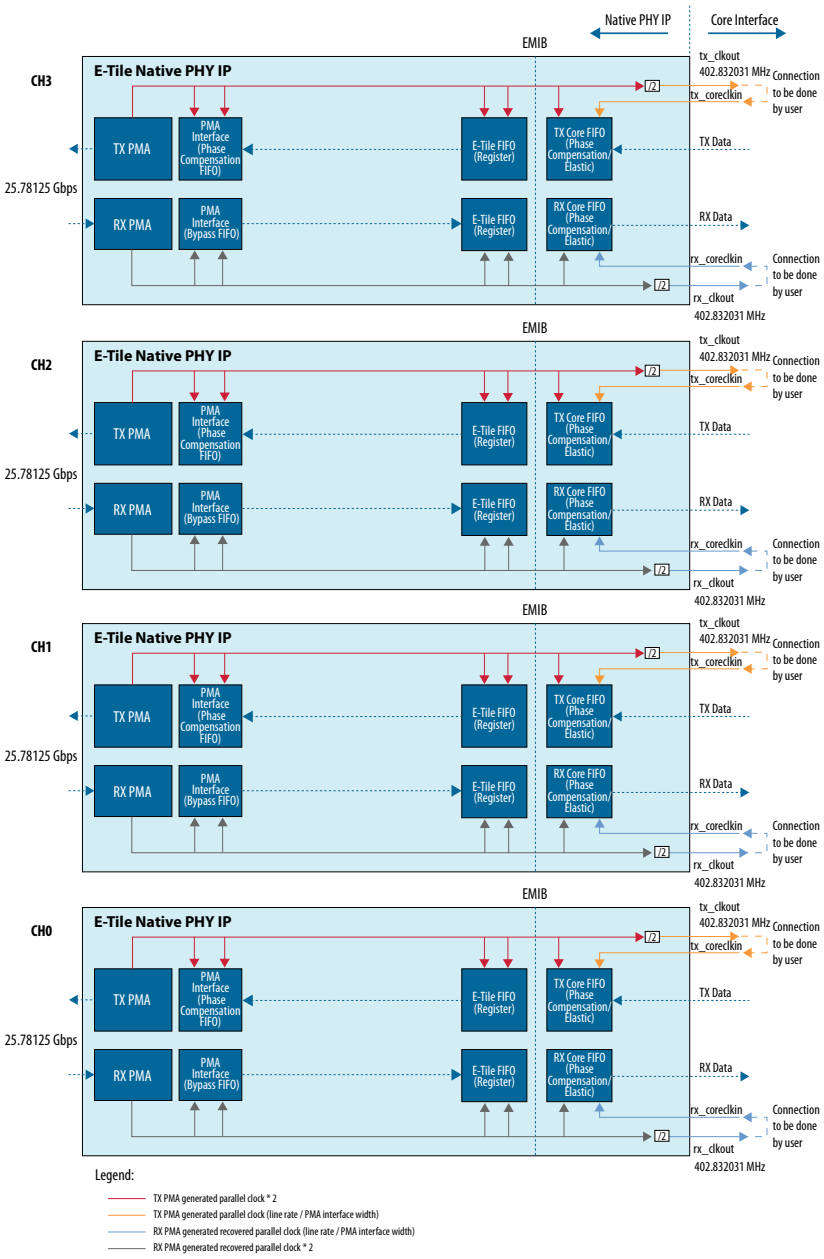
4.2.4. PMA Direct 25 Gbps x 4 (FEC Off)

This use case does not include FEC; therefore, there is no need for clock sharing between the four 25 Gbps channels.

For Core Interface FIFO in Phase Compensation mode, connect `tx_clkout` (402.832031 MHz) to `tx_coreclkin` and connect `rx_clkout` (402.832031 MHz) to `rx_coreclkin`. If you use any other source for `tx_coreclkin` or `rx_coreclkin`,

make sure `tx_coreclkin` and `rx_coreclkin` have 0 PPM difference with the `tx_clkout` and `rx_clkout`, respectively. This example assumes that TX and RX Double Width transfer is enabled.

Figure 69. PMA Direct 25 Gbps x 4 (FEC Off)



4.2.5. PMA Direct 10.3125 Gbps x 4

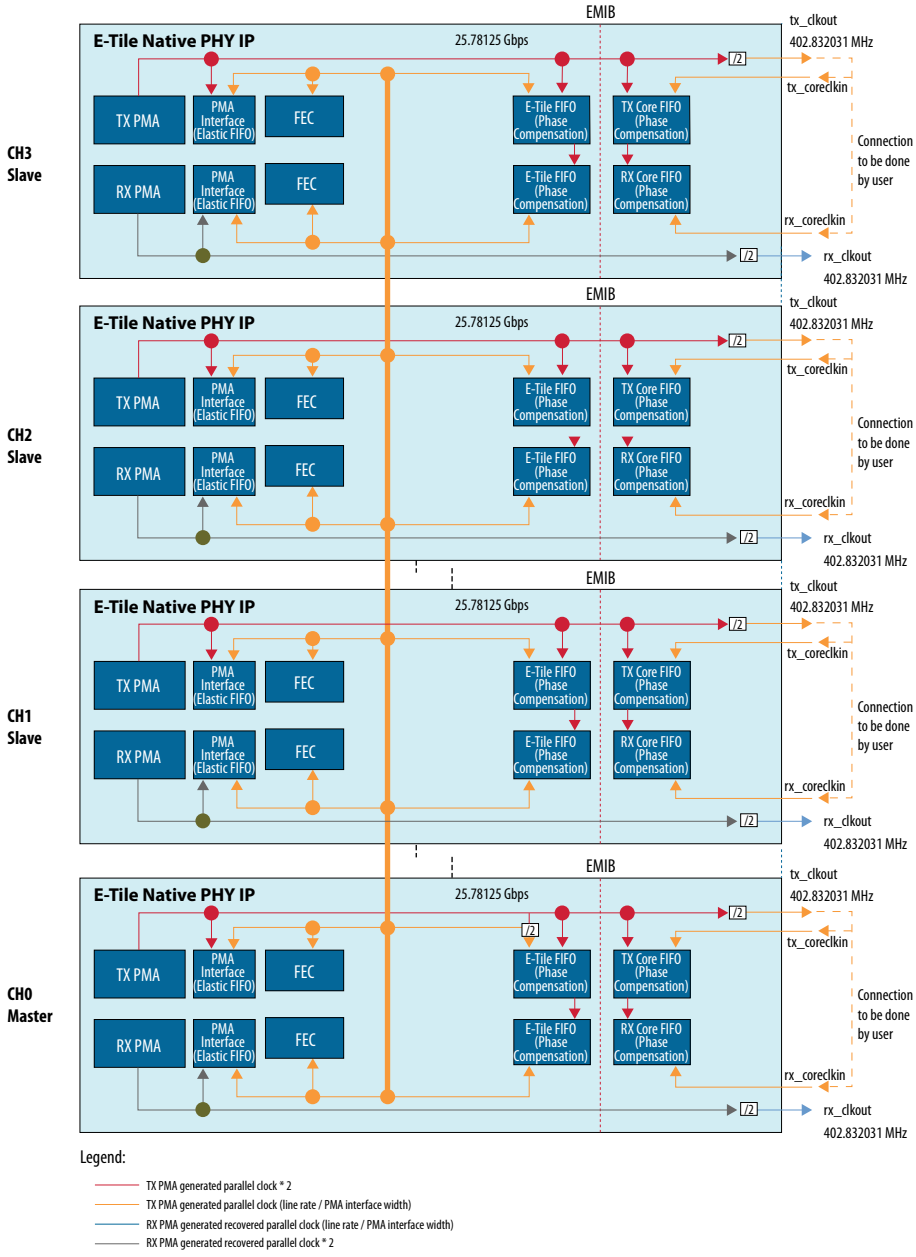
Connect `tx_clkout` (257.8125 MHz) of each individual channel to `tx_coreclkin` and connect `rx_clkout` (257.8125 MHz) to `rx_coreclkin`. If you use any other source for `tx_coreclkin`/`rx_coreclkin`, make sure `tx_coreclkin` and `rx_coreclkin` have 0 PPM difference with `tx_clkout` and `rx_clkout`, respectively. The clocking scheme in this use case is the same as Figure 66 on page 117.

4.2.6. PMA Direct 100GE Gbps (25 Gbps x 4) (FEC On)

This use case is implemented in the case of multi-lane protocols like 100GbE, for example. This uses four transceiver lanes of 25 Gbps each, where all four lanes must use the same FEC block. FEC is clocked by one of the four channels and you can configure this in the Native PHY IP core Parameter Editor. There is an inherent dependency between channels in this configuration. However, for applications like 100 GbE, dependency is acceptable and sometimes required. For each of the four channels with Core Interface FIFOs in Phase Compensation mode, connect `tx_clkout` (402.832031 MHz) to `tx_coreclkin` and `rx_coreclkin`. If you use any other source for `tx_coreclkin` or `rx_coreclkin`, make sure `tx_coreclkin` and `rx_coreclkin` have 0 PPM difference with the `tx_clkout`. The figure below assumes that all four channels have a common reference clock source (0 PPM between all four channels). You can also connect `tx_clkout` from any of the four channels to the `tx_coreclkin` and `rx_coreclkin` of all four channels. This helps keep the four-lane data synced on a single clock domain.

Figure 70. PMA Direct 100GE Gbps (25 Gbps x 4 per lane) (FEC On)

RS-FEC is also clocked by the TX PMA generated clock.



For clocking within the EHIP, see the *E-tile Hard IP for Ethernet Intel FPGA IP User Guide*.

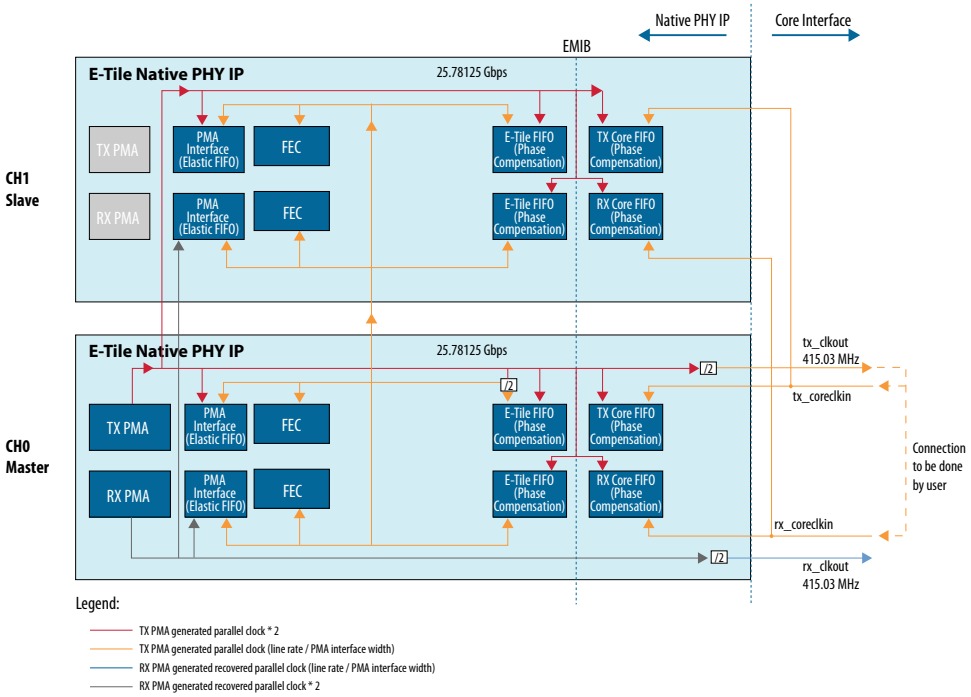
Related Information

E-tile Hard IP for Ethernet Intel FPGA IP User Guide

4.2.7. PMA Direct 100GE PAM4 (50 Gbps x 2) (Aggregate FEC On)

The figure below shows the clocking scheme for two channels out of four being used in a 100GE PAM4 (50 Gbps x 2) scheme. The FEC clock across all four channels is shared and driven by the master channel. For Core Interface FIFOs in Phase Compensation mode, connect half-rate `tx_clkout` (415.0390625 MHz) to `tx_coreclkin` and `rx_coreclkin`. If you use any other source for `tx_coreclkin`, make sure `tx_coreclkin` has 0 PPM difference with `tx_clkout`.

Figure 71. PMA Direct 100GE PAM4 (50 Gbps x 2) (Aggregate FEC On)



4.3. Clock Network Revision History

Document Version	Changes
2020.01.31	Made the following change: • Clarified some blocks in the *Core Clock Network Use Case* figures.
2019.10.11	Made the following changes: • Added clarifications to *Reference Clock Pins* and *QSF Assignments for Reference Clock Pins*. • Added the Related Information links for the Intel Agilex device documents.
2019.07.29	Made the following changes: • Added *Dynamic Reconfiguration of Reference Clock*. • Changed 402.83 to 402.832031.
2019.02.04	Made the following changes:

continued...

Document Version	Changes
	• Added instructions for connecting `tx_clkout`, `tx_coreclkin`, `rx_clkout`, and `rx_coreclkin`. • Clarified *Master-Slave Configuration: Option 2*. • Add recommendations for the *QSF Assignments for Reference Clock Pins*. • Changed the maximum reference clock frequency from 500 to 700 and added related instructions to *Reference Clock Pins*, and clarified that, although the E-tile transceiver reference clock input pin supports a frequency range of 125 MHz to 700 MHz, the reference clock network supports a maximum frequency of 500 MHz. • For *25 Gbps PMA Direct Channel (with FEC) within a Single FEC Block*, clarified that TX and RX Double Width is enabled.
2018.10.08	Made the following changes: • Updated the "Clock Sharing 25G Ethernet + 24G CPRI" figure. • Updated the "Clock Sharing 25G Ethernet + 24G CPRI + PMA Direct" figure. • Added *Use Cases* and all subsections. • Removed the "Clocking Sharing Across Multiple IPs" section. • Added the "E-Tile Channel Placement for a Single 25-Gbps PMA Direct Channel (with FEC) Within a Single FEC Block" figure. • Changed the QSF assignment for all parameters in the "QSF Assignments for a Single Reference Clock Pin (refclk[0])" table.
2018.07.18	Made the following changes: • Added *QSF Assignments for Reference Clock Pins*. • Added *Clocking Sharing Across Multiple IPs*.
2018.05.15	Made the following changes: • Updated figure "REFCLK LVPECL Pins" so that refclk_in_B only connects to REFCLK_1. • Changed REFCLK to "reference clock" except for references to REFCLK[0-8] and `refclk` pins.
2018.01.31	Initial release.

5. PMA Calibration

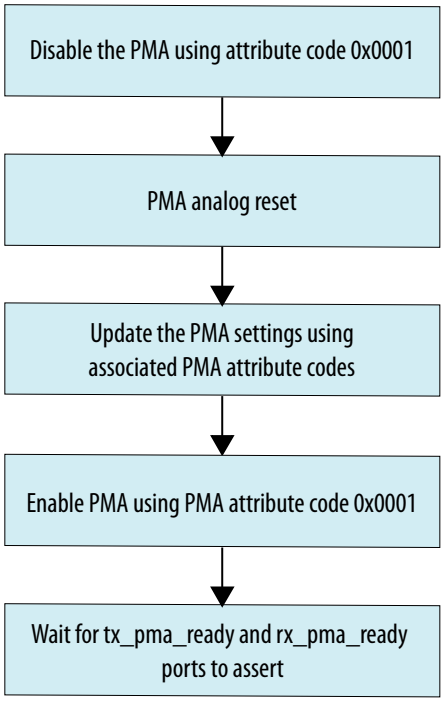
The PMA is automatically calibrated when it is first enabled. When dynamically reconfiguring the PMA to a different data rate, you need to recalibrate the PMA. The PMA calibration is run when the PMA is enabled after a PMA analog reset, so you perform a PMA analog reset before requesting calibration. Refer to *PMA Attribute Details* for information on how to change PMA settings using the AVMM interface. When referring to the *PMA Attribute Codes* section of the register map keep the following guidelines in mind:

- If you change any setting where **yes** is indicated in the **PMA Can be Running While Updating** column, ensure that you do not disable the transceiver channel. If you disable the channel, then the transition of disable to enable initiates the calibration and causes bit errors.
- If you change any setting where **no** is indicated in the **PMA Can be Running While Updating** column, follow the following flow chart.

Refer to *PMA Attribute Codes* for information on which attributes can only be changed when the PMA is disabled.

When the Native PHY's `tx_pma_ready/rx_pma_ready` outputs assert, calibration is complete.

Figure 72. Enabling PMA Calibration



Related Information

- PMA Attribute Details on page 161
- PMA Attribute Codes on page 216
- Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode on page 179

5.1. PMA Calibration Revision History

Document Version	Changes
2019.02.04	Made the following changes: • Added link to *Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode*. • Added reminder: Refer to *PMA Attribute Codes* for information on which attributes can only be changed when the PMA is disabled.
2018.07.18	Made the following changes: • Added instructions to *PMA Calibration*: When dynamically reconfiguring the PMA to a different data rate, you need to recalibrate the PMA. The PMA calibration is run when the PMA is enabled, so you first disable the PMA. • Added the "Wait for tx_pma_ready and rx_pma_ready ports to assert" block to the "Enabling PMA Calibration" figure.
2018.05.15	Made the following changes: • Added instructions to enable/disable the PMA using PMA attribute code 0x0001. • Added the "Changing a Setting When the PMA Cannot be Running" flowchart.
2018.01.31	Initial release.

6. Resetting Transceiver Channels

Intel recommends a reset sequence that ensures the physical medium attachment (PMA) in each transceiver channel initialize and function correctly.

6.1. When Is Reset Required?

You can reset the transmitter (TX) and receiver (RX) data paths independently or together. To ensure that transceiver channels are ready to transmit and receive data, you must properly reset the transceiver PHY after any of the following events.

Table 53. Digital Reset Conditions

Event	Reset Requirement
Device power-up and configuration	Requires the following resets which can be done in any order or at the same time • Requires toggling the TX and RX resets after power-up but before there is any traffic • Requires toggling `reconfig_reset` after power-up but before actual use of the NPDME, MIF streamer, PMA configuration, or control and status Avalon memory-mapped interface registers
Channel dynamic reconfiguration	Requires holding the TX and RX in analog and digital reset before reading or writing registers for the PMA that may cause a rate change
Any event requiring a PMA reset such as a reference clock changes or serial data loss	Requires a transceiver channel reset

6.2. How Do I Reset?

E-Tile transceivers have separate reset procedures for analog reset and digital reset.

You can use the PMA attribute code 0x0001 on the AVMM reconfiguration bus to enable or disable the PMA. Digital reset can be asserted using the digital reset controller in the Native PHY IP.

There are special reset procedures to follow if the E-Tile Native PHY IP core is configured with the RS-FEC enabled.

Table 54. Reset Requirements when RS-FEC is Enabled

Number of Channels	Enable RS-FEC	Enable Datapath and Interface	Aggregate Mode	Reset Controller
1 to 24	No	No	N/A	Manual or automatic mode
1 to 3	Yes	No	N/A	Bypassed
1 to 3	Yes	Yes	N/A	Bypassed
4	Yes	No	No	Bypassed

continued...

Number of Channels	Enable RS-FEC	Enable Datapath and Interface	Aggregate Mode	Reset Controller
4	Yes	Yes	No	Bypass
4	Yes	No	Yes	Manual or automatic mode
4	Yes	Yes	Yes	Bypassed

The reset tab does not appear when you bypass the reset controller. See *Reset Controller Bypass* for details about how to reset the transceiver digital logic when the reset controller is bypassed.

Note: When resetting both the analog and digital logic, the digital reset controller cannot reset the PMA. Therefore, you must monitor the `tx_pma_ready/rx_pma_ready` outputs and make sure the PMA has been reset successfully through the AVMM reconfiguration bus before deasserting the digital reset inputs. This ensures that the reset controller does not accidentally deassert reset to different sub-blocks like EMIB or RS-FEC before the PMA comes out of reset.

Related Information

Reset Controller Bypass on page 142

6.2.1. Disabling the E-Tile Transceiver

Use the steps below when both digital and analog resets are needed. Changing the data rate is an example of when both digital and analog resets are needed.

1. Assert the digital reset controller. See *High Level Specification* for the ports and sequence.
2. Wait for the Native PHY's `tx_ready/rx_ready` outputs to deassert.
3. Disable the PMA on the AVMM reconfiguration bus using PMA attribute code 0x0001. The PMA's TX, RX, or both TX and RX can be disabled.
4. Wait for the Native PHY IP's `tx_pma_ready/rx_pma_ready` output to deassert.
5. If conditions exist that require a *PMA Analog Reset*, assert the PMA analog reset then load the new PMA settings.
6. Enable the PMA using PMA attribute code 0x0001.
7. Wait for the Native PHY IP's `tx_pma_ready/rx_pma_ready` to assert.
8. Deassert the digital reset controller. See *High Level Specification* for the ports and sequence.

The `tx_pma_ready/rx_pma_ready` assert after the PMA finishes reconfiguration, followed by the `rx_is_lockedtodata` output. Finally, the `tx_ready/rx_ready` outputs assert to indicate the transceiver channels have finished reset.

Related Information

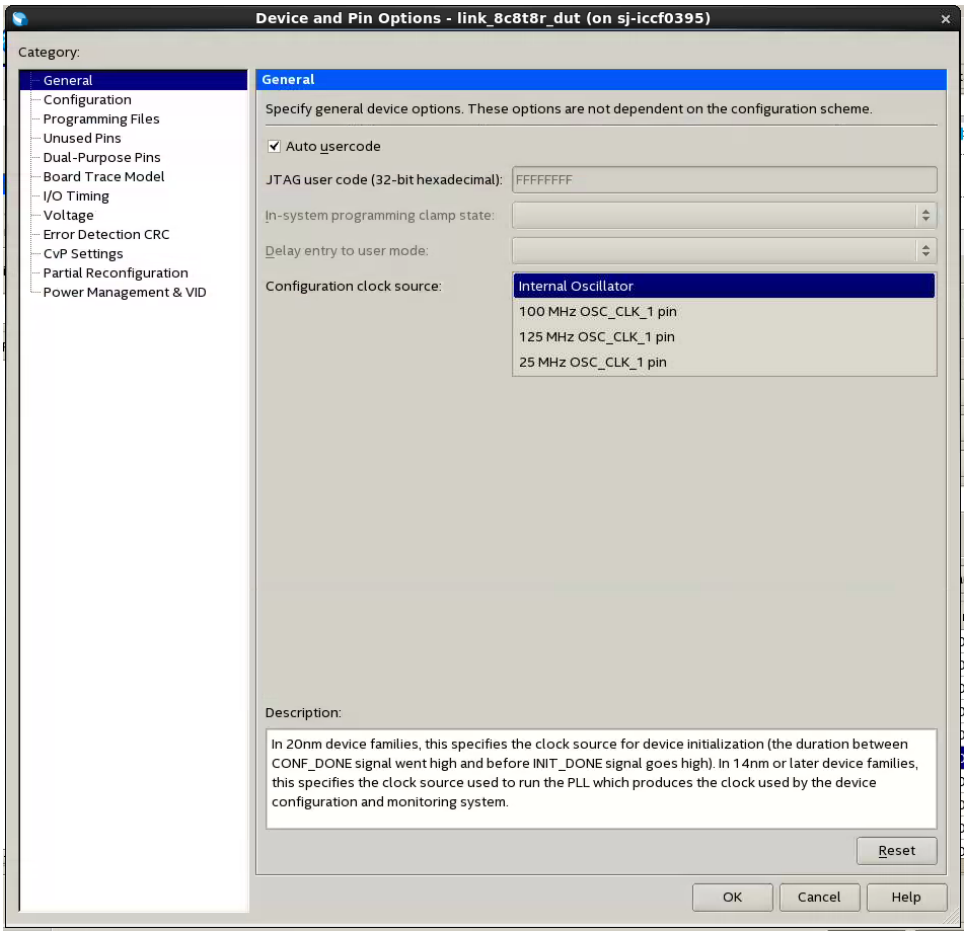
- High Level Specification on page 134
- PMA Analog Reset on page 132

6.2.2. Selecting the Reset Controller's Clock Source

When you instantiate a Native PHY IP, the software automatically instantiates Master Transceiver Reset Sequencer (TRS) and Local TRS (LTRS) blocks. Use the Intel Quartus Prime Pro Edition assignment settings editor to provide a 25, 100, or 125 MHz free-running and stable clock to OSC_CLK_1 for the proper functionality of the two blocks. To set the OSC_CLK_1 frequency in Intel Quartus Prime, follow these steps:

1. Select **Assignment ➤ Settings**.
2. Click **Device/Board** in the top right corner.
3. Select **Device and Pin Options**.
4. Select **25 MHz OSC_CLK_1 pin**, **100 MHz OSC_CLK_1 pin**, or **125 MHz OSC_CLK_1 pin** as the **Configuration clock source**.

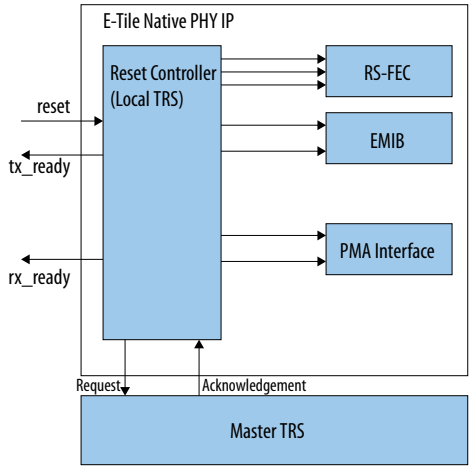
Figure 73. Device and Pin Options



6.3. Reset Block Architecture

The Native PHY IP core's digital reset controller block interacts with a master and local transceiver reset sequencer. The Master TRS and Local TRS blocks work together to stagger the resets to the transceiver channels for noise mitigation.

Figure 74. Reset Block Diagram with Single Reset Control



The Intel Quartus Prime Pro Edition software detects the presence of instantiated transceiver Native PHY IP cores and automatically inserts the TRS. The `tx_reset` and `rx_reset` inputs, either generated by you or through the reset controller, are received by the Local TRS. The Local TRS also forwards the request to the master TRS for scheduling. TRSs work together to schedule all the requested RS-FEC/PMAIF resets and provide acknowledgment for each request. Use either the reset controller inside the transceiver PHY or your own reset controller with the transceiver reset in manual mode. However, for the TRS to work correctly, the required timing duration must be followed.

Note: The master and local TRS IP is an inferred block and is not visible in the RTL. You have no control over this block.

Table 55. Reset Signals Required for E-Tile

Reset	Transceiver Reset	Category
`tx_reset`	TX EMIB reset	EMIB Reset
	TX PMAIF reset	Transceiver Interface Reset
	RS-FEC reset	RS-FEC Reset
	TX RS-FEC reset	General RS-FEC reset and includes the TX and RX datapath
`rx_reset`	RX EMIB reset	EMIB Reset
	RX PMAIF reset	Transceiver Interface Reset
	RX RS-FEC reset	RS-FEC Reset on RX datapath

The `tx_reset` and `rx_reset` signals apply the associated transceiver resets.

You can use the Native PHY's AVMM interface to do a PMA analog reset or to enable and disable the PMA.

You have the option to use `tx_reset` and `rx_reset` as the input controls if you enable independent TX and RX reset, or you can use `reset` as the input to control both TX and RX if you disable independent TX and RX reset. The diagrams "Reset

Block Diagram with Independent TX and RX Reset Control" and "Reset Block Diagram with Single Reset Control" in *Automatic Reset Mode* show the reset IP in both conditions.

You can use the reset controller in automatic or manual reset mode for PMA direct modes, but you need to use the reset controller bypass when using the RS-FEC block in fractured mode or if you want to reconfigure from RS-FEC On to RS-FEC Off or from RS-FEC Off to RS-FEC On.

Resets signals for the Ethernet Hard IP are not included. See *E-tile Hard IP for Ethernet Intel FPGA IP User Guide* for details.

Related Information

- High Level Specification on page 134
- Automatic Reset Mode on page 134
- *E-tile Hard IP for Ethernet Intel FPGA IP User Guide*

6.4. PMA Analog Reset

The transceiver has internal logic that is clocked by the transceiver's reference clock. There is only one PMA reset, and it affects both TX and RX PMAs.

Table 56. When to Perform a PMA Analog Reset

Event	Reason to Reset
Powering up	Initial adaptation does not help bring up the device in a good state.
Correcting a loose cable connection and bringing up a link	Initial adaptation does not help bring up the device in a good state.
Changing the reference clock frequency or reference clock source	Dynamic reconfiguration does not operate properly. See *Dynamic Reconfiguration* for details.
Changing data rates (both transceiver and link)	Dynamic reconfiguration does not operate properly. See *Dynamic Reconfiguration* for details.
Changing channel physical connections (for example, going from cable to backplane)	Initial adaptation does not help bring up the device in a good state.
Changing or interrupting the transceiver power supply	Initial adaptation does not help bring up the device in a good state.

To reset the internal controller, use the transceiver's AVMM bus to:

1. Write 0x200[7:0] = 0x00.
2. Write 0x201[7:0] = 0x00.
3. Write 0x202[7:0] = 0x00.
4. Write 0x203[7:0] = 0x81.

5. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.
6. Read-modify-write 0x95[5] = 0x1 to enable PMA calibration when loading the new PMA settings.
7. Resetting the internal logic causes the transceiver to lose its settings for the data rate and serialization/deserialization values. Write 0x91[0] = 1 to load the initial settings in the programming file (when the embedded MIF streamer is not used), or go to the last profile (when the embedded MIF streamer is used).

Alternatively, you can load in a new profile using the MIF streamer or by loading in the PMA settings by sending the individual attributes. See *Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode* for an example. Loading the PMA settings using register 0x91[0] or the MIF streamer enables the PMA.

Table 57. Alternatives to a PMA Analog Reset

Use Case	Description	Solution
Link fault management or fault handling of a link-down	Systems relying on IEEE link fault handling are impacted by a PMA analog reset bringing down both the transmitter and receiver. The transmitter is turned off after a PMA analog reset which triggers a local fault on the link partner. The receiver needs to recover from the loss of a lock without a PMA analog reset. Turning on or off transmitter and receiver buffers and the recalibrating the receiver helps narrow down the issue during link-down scenarios.	Enable or disable the TX and RX buffers (see *0x0001: PMA Enable/Disable*). Use *Initializing an RX* and "Initial Adaptation" in *PMA Receiver Equalization Adaptation Usage Model* to recalibrate the receiver.
Turning on internal serial loopback from mission mode	A PMA analog reset can be avoided by resetting the receiver equalization.	Use *Resetting the RX Equalization*.
Changing the loss profile for a given medium	Changing media from short reach to long reach may require a PMA analog reset.	Use *Initializing an RX* and "Initial Adaptation" in *PMA Receiver Equalization Adaptation Usage Model* to recalibrate the receiver.
Handling different link partners for the transmitter and receiver of a given channel	This requires independent receiver adaptation without a PMA analog reset so that the transmitter is not disrupted.	Use *Initializing an RX* and "Initial Adaptation" in *PMA Receiver Equalization Adaptation Usage Model* to recalibrate the receiver.
Speed training for CPRI	Turning off the transmitter buffer during link training avoids the need for a transmitter reset or PMA analog reset.	Enable or disable the TX and RX buffers (see *0x0001: PMA Enable/Disable*).

Related Information

- Dynamic Reconfiguration on page 150
- Switching Reference Clocks on page 161
- Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode on page 179
- PMA Analog Reset on page 237
- 0x0001: PMA Enable/Disable on page 217
- PMA Receiver Equalization Adaptation Usage Model on page 188
- Initializing an RX on page 206
- Resetting the RX Equalization on page 209

6.5. High Level Specification

The overall E-tile reset sequencing solution consists of Reset Controller, a Master Transceiver Reset Sequencer (Master TRS) and multiple Local Transceiver Reset Sequencers (Local TRS or LTRS).

The Reset Controller is used to ensure proper timing requirements and interconnections. It takes the reset signals and handles the assertion and deassertion of TX reset and RX reset to Local TRS. It also gives the option to enable and disable independent TX and RX reset, enable and disable independent channel reset and provides fast simulation support.

A Master TRS services the reset requests from multiple Local TRS in a round-robin fashion. Upon detecting a reset event (assertion or deassertion) on the reset signal, the Local TRS raises a reset request to the Master TRS and waits for the reset acknowledgment from the Master TRS. When it receives acknowledgment, the Local TRS sends the reset event to the transceiver channels. If required, the Local TRS sequences the actual reset signals that go to the channels and adds extra delays to the reset assertion or deassertion. After the Local TRS is done with the reset, it drops the reset request; then the Master TRS moves to the next Local TRS request.

There are two variants of Local TRS – TX LTRS and RX LTRS. The TX LTRS services the TX reset signal, and the RX LTRS services RX reset signal. The Master ensures that a characterized minimum separation time is honored between the reset acknowledgments of any two reset requests across the device, ensuring the minimum separation time between the reset events of any two reset signals across the device. The Local TRS provides more delay if required by the reset signal.

6.5.1. Automatic Reset Mode

Figure 75. Reset Block Diagram with Single Reset Control

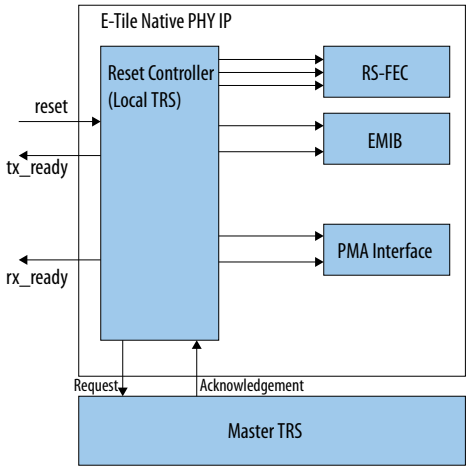


Figure 76. Reset Block Diagram with Independent TX and RX Reset Controls

The reset controller can be subdivided into TX and RX reset controllers. This allows you to reset the TX or RX independently.

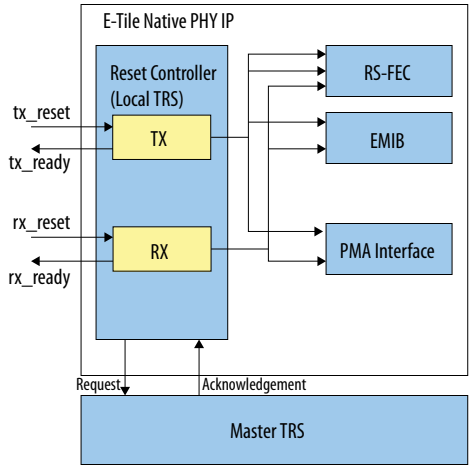
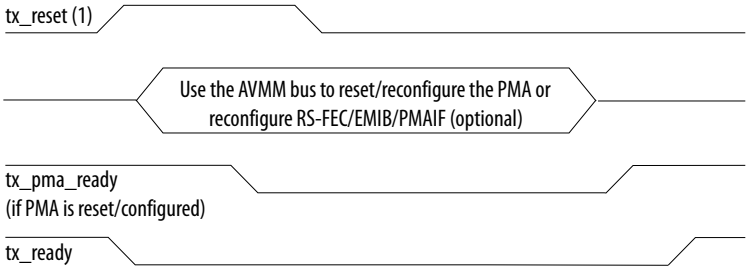
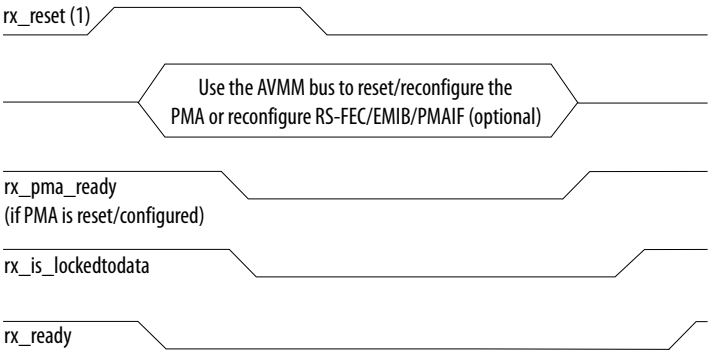


Figure 77. TX Reset Sequence in Automatic Mode After Power-Up



Note:
1. If you used the AVMM bus to reconfigure the RS-FEC/EMIB/PMAIF, you must assert tx_reset until the RS-FEC/EMIB/PMAIF registers are written.

Figure 78. RX Reset Sequence in Automatic Mode After Power-Up



Note:
1. If you used the AVMM bus to reconfigure the RS-FEC/EMIB/PMAIF, you must assert rx_reset until the RS-FEC/EMIB/PMAIF registers are written.

6.5.2. Manual Reset Mode

In manual mode, all ports are exposed to provide flexible control. Follow the reset sequence for RX and TX modes to send reset requests.

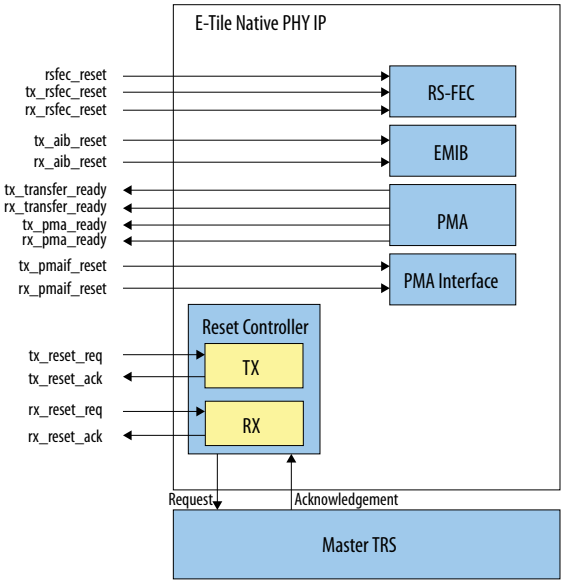
Note: The manual reset mode is required if fractured RS-FEC is used.

Table 58. Native PHY IP Ports With Manual Mode Enabled

Port	Direction	Description
rx_reset_req	Input	Request to Master TRS to schedule RX reset
rx_reset_ack	Output	Valid window for you to assert/deassert rx_aib_reset, rx_pmaif_reset, rx_rsfec_reset
rx_aib_reset	Input	Reset RX EMIB datapath
rx_pmaif_reset	Input	Reset RX PMA digital logic
rx_rsfec_reset	Input	Reset RX RS-FEC datapath
rx_transfer_ready	Output	Output from the Native PHY IP core indicating the RX EMIB datapath is ready
rx_pma_ready	Output	Output from the PMA indicating the PMA is ready. This must be asserted before asserting or deasserting any RX resets.
rx_is_lockedtodata	Output	Output from the PMA indicating the CDR has locked to the incoming serial data
tx_reset_req	Input	Request to Master TRS to schedule TX reset
tx_reset_ack	Output	Valid window to assert or deassert tx_aib_reset, tx_pmaif_reset, tx_rsfec_reset, rsfec_reset
rsfec_reset	Input	Reset all RS-FEC logic
tx_aib_reset	Input	Reset TX EMIB datapath
tx_pmaif_reset	Input	Reset TX PMA digital logic
tx_rsfec_reset	Input	Reset TX RS-FEC datapath
tx_transfer_ready	Output	Output from the Native PHY IP core indicating the TX EMIB datapath is ready
tx_pma_ready	Output	Output from the PMA indicating the PMA is ready. This must be asserted before asserting or deasserting any TX resets.

The reset, rx_ready, and tx_ready ports do not appear in manual reset mode.

Figure 79. Manual Reset Mode



You assert the `tx_reset_req` or `rx_reset_req` ports to start the digital reset process. You need to assert `tx_reset_req` or `rx_reset_req` every time you want to assert or deassert reset signals. You can assert `req` ports on multiple channels at the same time. The Local TRS and Master TRS round robin and stagger the resets. However:

- If you use the RS-FEC block and want to reset both the TX and RX, you must complete the TX reset on a specific channel before resetting the RX on that channel.
- You must ensure that the `tx_pma_ready` output is asserted before asserting the `tx_reset_req`.
- You must ensure that the `rx_pma_ready` output is asserted before asserting the `rx_reset_req`.
- You must monitor `rx_is_lockedtodata`.
- After `rx_lockedtodata` stays high for 180 µs, you may deassert the RX digital resets.

The following use model is supported:

1. You assert multiple `reset_req`. The Local TRS forwards the reset_req signal to the Master TRS.
2. The Master TRS selects one of the `reset_req` and waits 200 ns before asserting the reset_ack output.
3. You assert the resets on the EMIB, RS-FEC, and PMA interfaces. See Figure 81 on page 138 through Figure 86 on page 141 for TX and RX reset sequences.
4. You deassert the `reset_req` signal after resetting the blocks.
5. The Master TRS sees the deasserted `reset_req` and deasserts the `reset_ack` output.

Note: The Master TRS automatically deasserts the `reset_ack` output after 400 µs if you have not deasserted the `reset_req` input. In that case, you must deassert and reassert the `reset_req` input to enter the round robin pool again.

6. The Master TRS goes to the next request in a round robin fashion and waits 200 ns before asserting the next `reset_ack`.

The figure below shows how to use the `tx_reset_req/rx_reset_req` inputs to request a reset window and how `tx_reset_ack/rx_reset_ack` marks the Master TRS returning a valid reset window.

Figure 80. Manual Mode Reset Timing Model

During the timing window when the `reset_ack` output is high, reset the blocks in sequence. The numbers refer to the steps above.

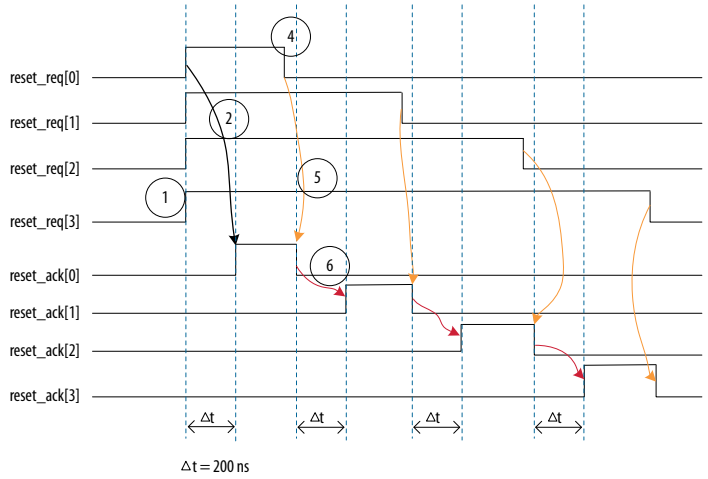
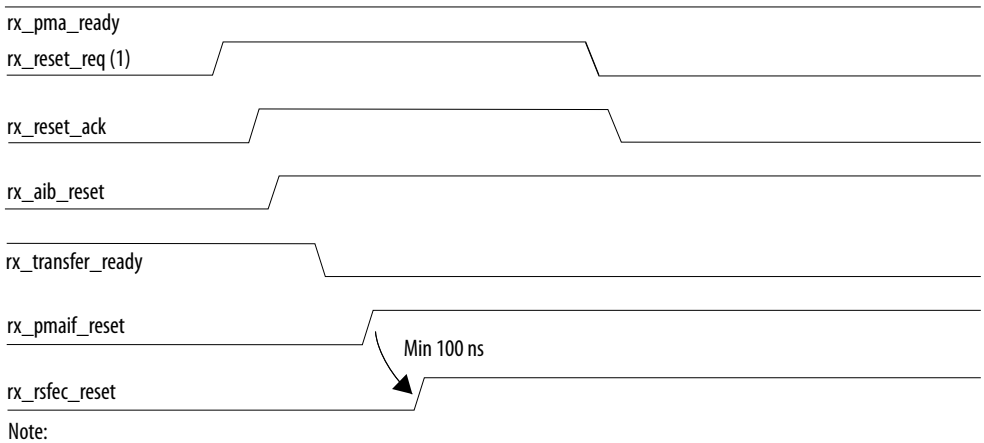


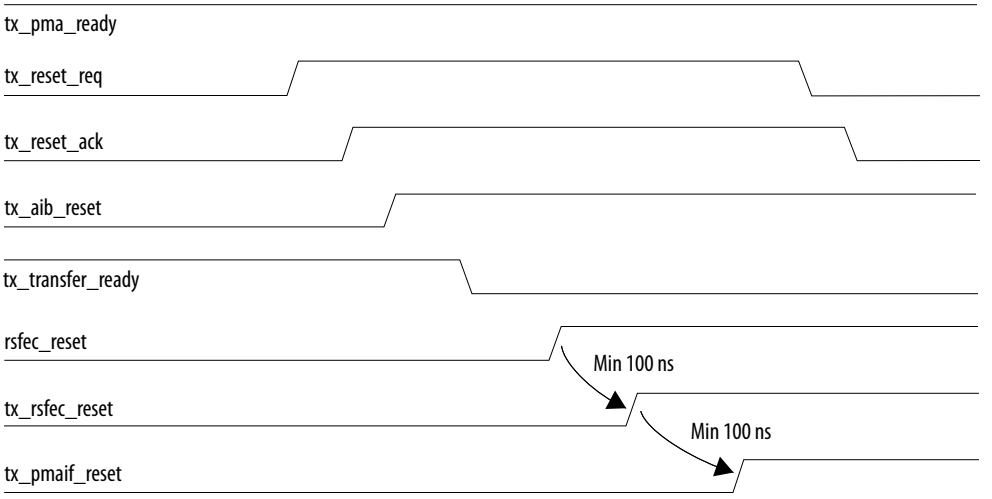
Figure 81 on page 138 and Figure 82 on page 139 below show how to assert TX and RX reset.

Figure 81. RX Reset Assertion Timing Waveform



Note:
1. If you enabled the RS-FEC block, you must assert rx_reset_req after the tx_transfer_ready output is asserted.
2. See Master-Slave Clocking Option 2 Reset Details for information about whether or not to assert tx_aib_reset or rx_aib_reset ports.

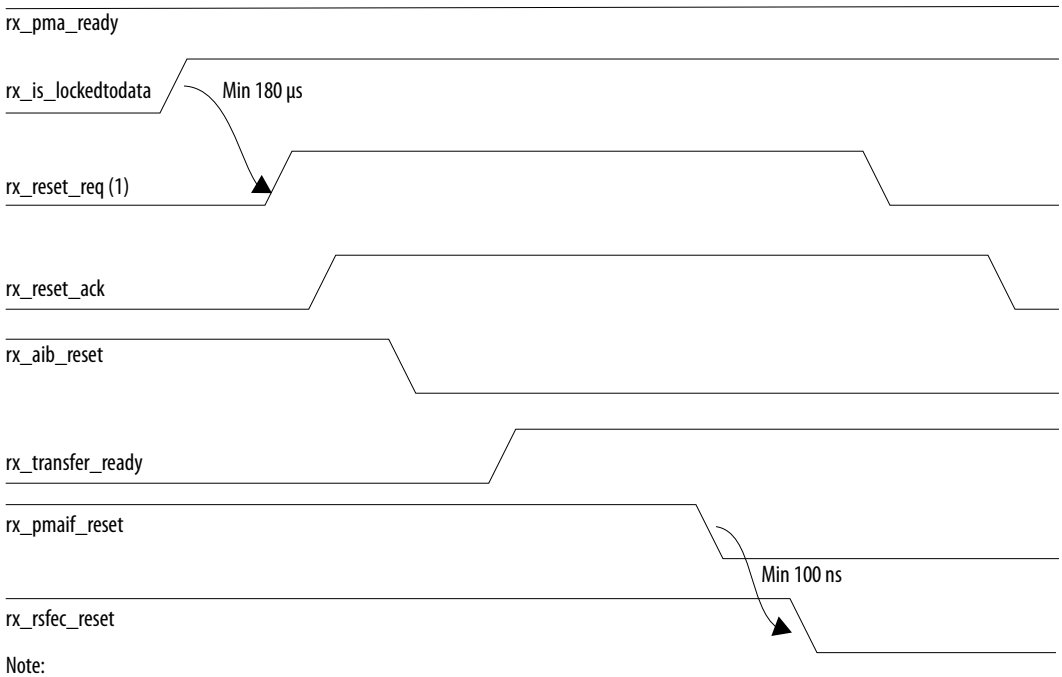
Figure 82. TX Reset Assertion Timing Waveform



Note:
1. See Master-Slave Clocking Option 2 Reset Details for information about whether or not to assert tx_aib_reset or rx_aib_reset ports.

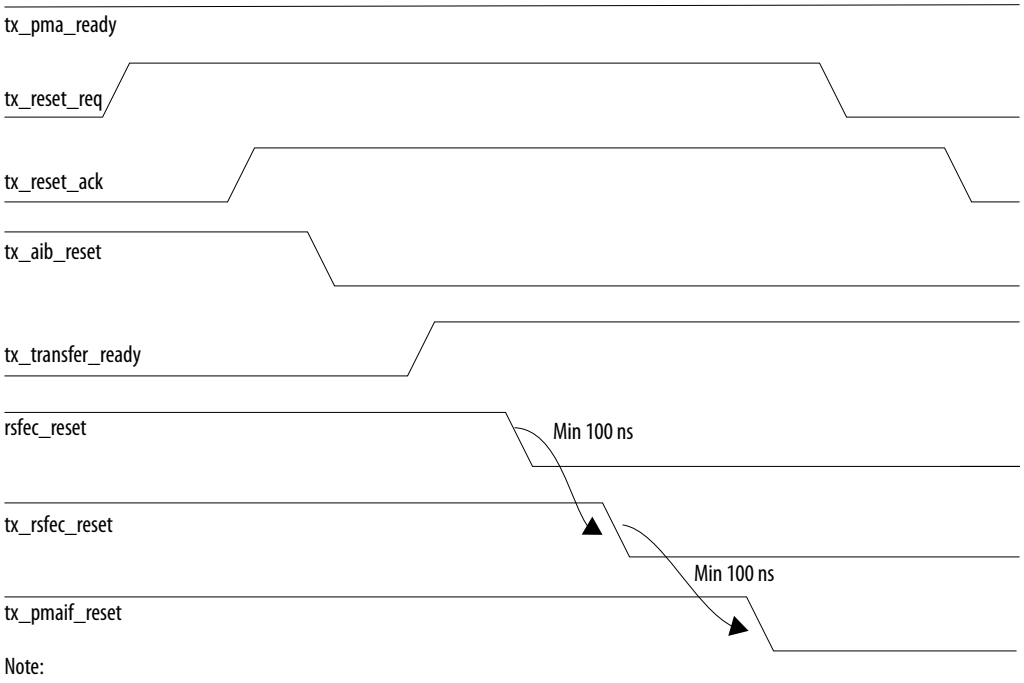
Figure 83 on page 139 and Figure 84 on page 140 below show how to deassert TX and RX reset.

Figure 83. RX Reset Deassertion Timing Waveform



Note:
1. If you enabled the RS-FEC block, you must assert rx_reset_req after the tx_transfer_ready output is asserted.
2. See Master-Slave Clocking Option 2 Reset Details for information about whether or not to assert tx_aib_reset or rx_aib_reset ports.

Figure 84. TX Reset Deassertion Timing Waveform

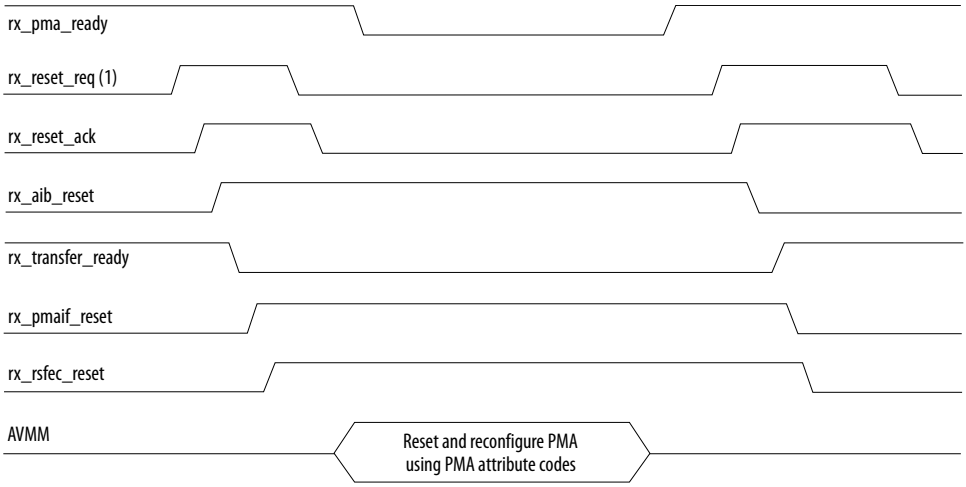


Note:
1. See Master-Slave Clocking Option 2 Reset Details for information about whether or not to assert tx_aib_reset or rx_aib_reset ports.

Because you only have 400 µs to complete a reset sequence, there is not enough time to assert the reset, reconfigure the PMA, and deassert the reset. So you should assert the reset in one reset window, reconfigure the PMA, and then deassert the reset in a second window. Refer to Figure 85 on page 141 and Figure 86 on page 141 below for details.

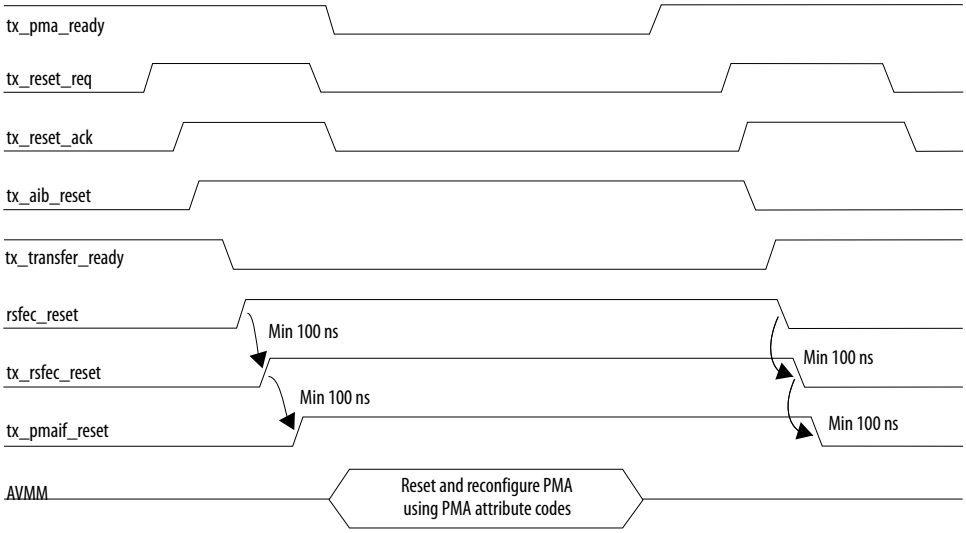
The RS-FEC block automatically locks onto the FEC symbols and you do not need to reset the RS-FEC block through the `rsfec_reset`, `tx_rsfec_reset`, or `rx_rsfec_reset` signals.

Figure 85. RX PMA Reconfiguration with Reset Controller in Manual Mode Timing Waveform



Note:
1. If you enabled the RS-FEC block, you must assert rx_reset_req after the tx_transfer_ready output is asserted.
2. See Master-Slave Clocking Option 2 Reset Details for information about whether or not to assert tx_aib_reset or rx_aib_reset ports.

Figure 86. TX PMA Reconfiguration with Reset Controller in Manual Mode Timing Waveform



Note:
1. See Master-Slave Clocking Option 2 Reset Details for information about whether or not to assert tx_aib_reset or rx_aib_reset ports.

Related Information

- Master-Slave Configuration: Option 2 on page 119
- Master-Slave Clocking Option 2 Reset Details on page 144

6.5.3. Reset Controller Bypass

You can access the resets for the internal PMA interface, RS-FEC, and EMIB blocks when you bypass the reset controller, much like when the reset controller is in manual mode.

The TRS block in the reset controller, which prevents multiple transceivers from being in reset at the same time is not implemented.

If you have multiple E-tile Native PHY IP core instances on a single E-tile, make sure that you assert/deassert reset to a single transceiver channel in an E-tile at a time. For example, if you instantiate three E-tile Native PHY IP cores with the following configurations:

- Instance A with one transceiver channel with RS-FEC disabled and the reset controller in automatic mode
- Instance B with four transceiver channels with RS-FEC enabled in aggregate mode and the reset controller in manual mode
- Instance C with two transceiver channels with RS-FEC enabled in fractured mode and the reset controller bypassed

If you want to reset instances A or B, you cannot assert/deassert reset signals on instance C at the same time.

If you want to reset instance C, you cannot reset instance A or B. If you previously asserted reset on instance A, ensure that `tx_ready` and `rx_ready` are deasserted on instance A. If you previously deasserted reset on instance A, ensure that `tx_ready` and `rx_ready` are asserted on instance A. Do not assert the `tx_reset_req` or `rx_reset_req` on instance B.

6.5.3.1. Reset Controller Bypass Ports

You can control the reset signals listed in the following table.

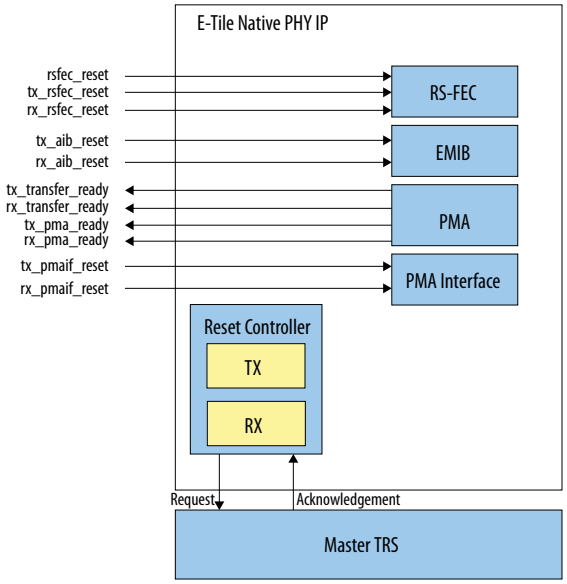
Table 59. Reset Controller Bypass Ports

Port	Input/Output	Description
`rx_aib_reset`	Input	Resets the RX EMIB datapath
`rx_pmaif_reset`	Input	Resets the RX PMA digital logic
`rx_rsfec_reset`	Input	Resets the RX RS-FEC datapath
`rx_transfer_ready`	Output	Output from the Native PHY IP core indicating the RX EMIB datapath is ready
`rx_pma_ready`	Output	Output from the PMA indicating the RX PMA is ready
`rx_is_lockedtodata`	Output	Output from the Native PHY indicating RX CDR is locked
`rsfec_reset`	Input	Resets all (TX and RX) RS-FEC logic
`tx_aib_reset`	Input	Resets the TX EMIB datapath
`tx_pmaif_reset`	Input	Resets the TX PMA digital logic

continued...

Port	Input/Output	Description
`tx_rsfec_reset`	Input	Resets the TX RS-FEC datapath
`tx_transfer_ready`	Output	Output from the Native PHY IP core indicating the TX EMIB datapath is ready
`tx_PMA_ready`	Output	Output from the PMA indicating the TX PMA is ready. This must be asserted before asserting or deasserting any TX resets

Figure 87. Reset Controller Bypass Ports

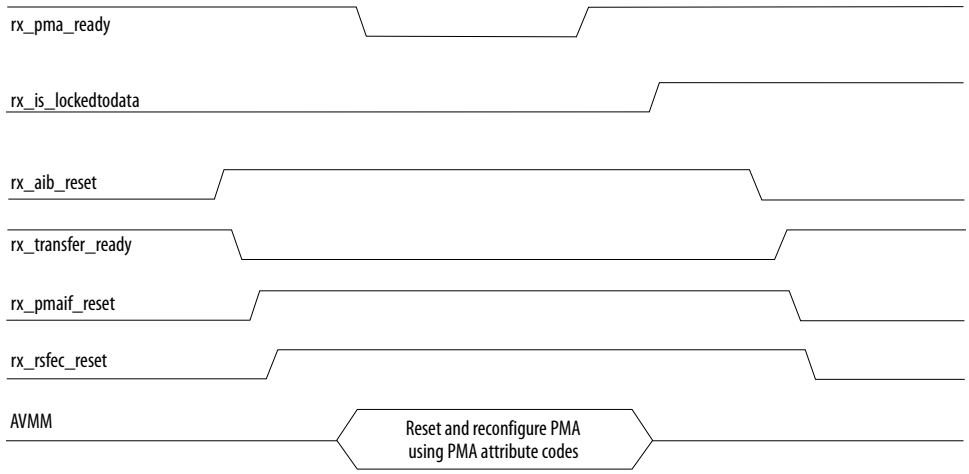


6.5.3.2. Reset Controller Bypass Reset Procedure

In Reset Controller Bypass mode, the reset controller is bypassed and therefore the local TRS and master TRS blocks are not implemented to circulate and stagger the resets.

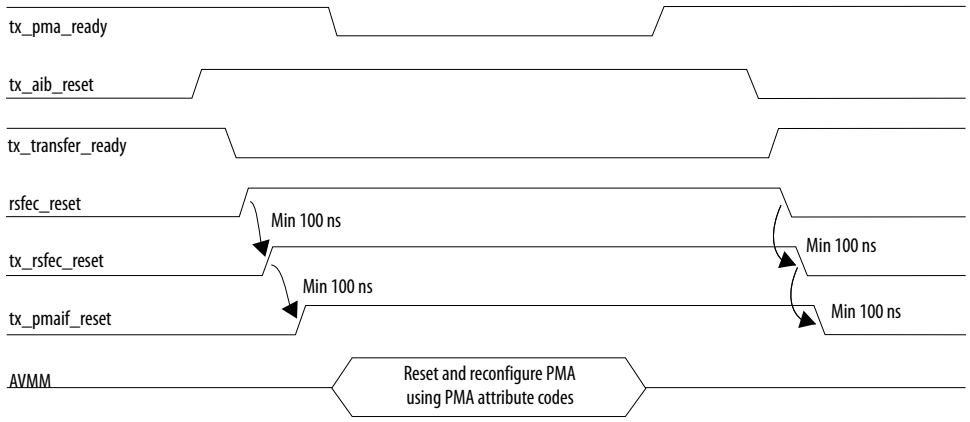
Because the RS-FEC is enabled, you must complete the TX reset on a specific channel before resetting the RX on that channel. Ensure the PMA is ready before asserting or deasserting reset to the individual transceiver digital blocks. Ensure the `tx_pma_ready` output is asserted before asserting/deasserting the `tx_pmaif_reset`, `tx_rsfec_reset`, `tx_aib_reset`, or `rsfec_reset` inputs. Ensure that you assert the `rx_pma_ready` output before asserting or deasserting the `rx_pmaif_reset`, `rx_rsfec_reset`, or `rx_aib_reset` inputs.

Figure 88. RX Reset Timing Waveform



Note:
1. See Master-Slave Clocking Option 2 Reset Details for information about whether or not to assert tx_aib_reset or rx_aib_reset ports.

Figure 89. TX Reset Timing Waveform



Note:
1. See Master-Slave Clocking Option 2 Reset Details for information about whether or not to assert tx_aib_reset or rx_aib_reset ports.

Related Information

- Master-Slave Configuration: Option 2 on page 119
- Master-Slave Clocking Option 2 Reset Details on page 144

6.6. Master-Slave Clocking Option 2 Reset Details

The following table summarizes whether to assert `tx_aib_reset` or `rx_aib_reset` for different reset conditions when in master-slave clocking option 2.

When multiple transceiver channels are configured in master-slave clocking option 2 (an external transceiver channel configured as a PLL generates the clocks used to transfer data across the EMIB), one channel acts as the master channel. Asserting `tx_aib_reset` or `rx_aib_reset` ports, visible in manual reset or reset controller bypass modes, on the master channel causes the EMIB transfer clock from the external PLL to not be propagated to the slave channels. This results in both the

master and slave channels to stopping the transfer of data across the EMIB. EMIB reset is only required when the clock used to transfer data across the EMIB gets disrupted, which does not occur in master-slave clocking option 2.

Additionally, the master channel cannot be dynamically reconfigured from RS-FEC enabled mode to PMA direct mode.

Table 60. Whether or Not to Assert `tx_aib_reset` or `rx_aib_reset` Ports Based on Reset Conditions

Channel type	Dynamic reconfiguration without data rate change	Dynamic reconfiguration with data rate change	Dynamic reconfiguration from RS-FEC enabled to PMA direct
Master	Do not assert `tx_aib_reset` or `rx_aib_reset`	Do not assert `tx_aib_reset` or `rx_aib_reset`	Not supported
Slave	Do not assert `tx_aib_reset` or `rx_aib_reset`	Do not assert `tx_aib_reset` or `rx_aib_reset`	Assert `tx_aib_reset` and `rx_aib_reset`

6.7. Intel Quartus Prime Instantiated Transceiver Reset Sequencer

Intel Quartus Prime auto-infers the Master TRS during synthesis and auto-connects the Master TRS to the Local TRS using the debug fabric master end-point to slave end-point auto-connect technology.

Benefits

No intervention – The designer does not need to expose every reset request and acknowledge port on the interfaces of the design modules to be connected to the Master TRS.

Design modularity – Local changes to the number of transceiver reset signals at a lower hierarchy in a module does not require a chain of interface changes up to the Master TRS hierarchy, especially if the transceiver instance is deep down in the design hierarchy.

Tradeoffs

It is harder to debug a possible connectivity issue in Synthesis than debugging the RTL.

Any issue with the instantiation and connectivity needs to be fixed in Synthesis instead of in the design.

6.8. Block Diagrams

Figure 90. General Block Diagram for Reset Controller when Use Separate TX/RX Reset Per Channel is Turned ON and Enable Individual TX and RX Reset is Turned OFF

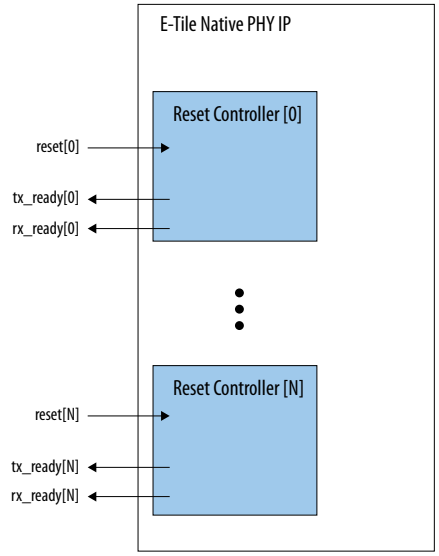


Figure 91. Reset Controller when Use Separate TX/RX Reset Per Channel is Turned ON and Enable Individual TX and RX Reset is Turned ON

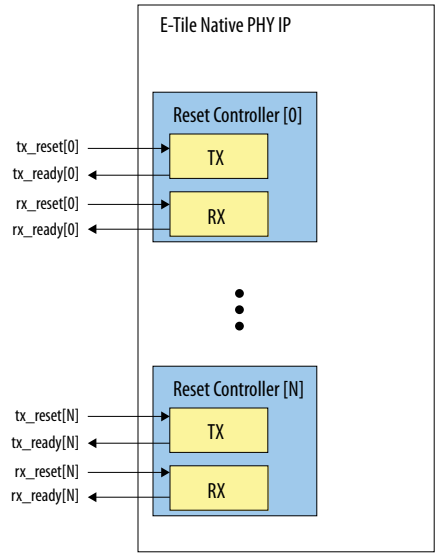
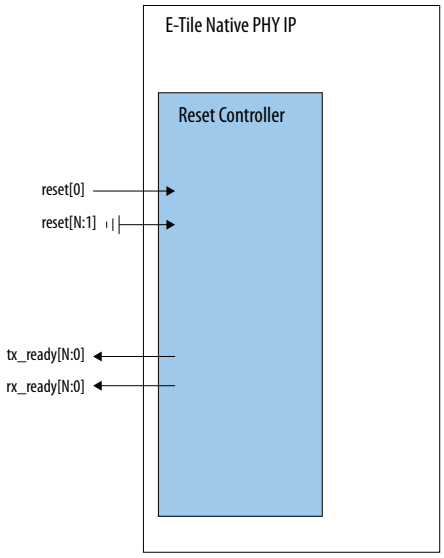


Figure 92. Reset Controller when Use Separate TX/RX Reset Per Channel is Turned OFF and Enable Individual TX and RX Reset is Turned OFF



6.9. Interfaces

6.9.1. Reset Parameters in the Native PHY GUI

Table 61. Reset Parameters

Parameter Name	Range	Description
Enable Manual Reset	On/Off	On = Use the manual reset process to reset the Native PHY.
Enable Fast Simulation for Controller	On/Off	On = Use reduced reset timings for simulation.
Enable Fast Simulation for Sequencer	On/Off	On = Use reduced reset timings for simulation.
Enable Individual TX and RX Reset	On/Off	On = Use individual reset to reset the TX or RX PMA path individually.
Use Separate TX/RX Reset Per Channel	On/Off	If you implement multiple channels in your Native PHY IP • On = Each channel's signals are independent from others and do not gate other channels' reset sequences. • Off = If any channel's PMA is not ready, deasserting `rx_is_lockedtodata` affects all channels. If you implement a single channel in the Native PHY IP, this has no effect.
Enable TX/RX reset sequencing	On/Off	On = The reset controller deasserts the TX before deasserting the RX. This parameter needs to be set when RS-FEC is enabled in aggregate mode and enable datapath and interface reconfiguration is disabled.
RX Reset Duration	100 ns to 1 ms	The RX pulses the `rx_is_lockedtodata` output when there is no serial data. The reset controller by default waits for the `rx_is_lockedtodata` to be high for 180 µs before deasserting reset. You can set this parameter for additional filtering.

6.9.2. HDL Ports/Interfaces

Table 62. HDL Ports/Interfaces when the Reset Controller is in Automatic Mode

Port Name	Direction	Width	Description
reset	Input	Number of channels	Resets TX and RX when asserted. Visible when **Enable individual TX and RX resets** is disabled. When the Native PHY is configured in PAM4 high data rate mode, the bus width equals the number of data channels parameter in the GUI divided by two.
tx_reset	Input	Number of channels	Resets TX when asserted. Visible when **Enable individual TX and RX resets** is enabled. When the Native PHY is configured in PAM4 high data rate mode, the bus width equals the number of data channels parameter in the GUI divided by two.
rx_reset	Input	Number of channels	Resets RX when asserted. Visible when **Enable individual TX and RX resets** is enabled. When the Native PHY is configured in PAM4 high data rate mode, the bus width equals the number of data channels parameter in the GUI divided by two.
tx_ready	Output	Number of channels	Status signal to indicate when TX resets sequencing is complete. Deasserts during TX reset assertion. Asserts a few clock cycles after deassertion of TX resets.
rx_ready	Output	Number of channels	Status signal to indicate when RX resets sequencing is complete. Deasserts during RX reset assertion. Asserts a few clock cycles after deassertion of RX resets.
tx_pma_ready	Output	Number of channels	One per channel. Indicates transceiver channel Transmit calibration completed.
rx_pma_ready	Output	Number of channels	One per channel. Indicates transceiver channel CDR calibration completed.

6.10. Resetting Transceiver Channels Revision History

Document Version	Changes
2020.01.31	Made the following changes: • Updated the "Digital Reset Conditions" requirements for device power-up and configuration. • Added the "Alternatives to a PMA Analog Reset" table to *PMA Analog Reset*.
2019.07.29	Made the following changes: • Added four actions to the "When to Perform a PMA Analog Reset" table. • Added *Master-Slave Clocking Option 2 Reset Details*. • Updated *Disabling the E-Tile Transceiver*.
2019.02.04	Made the following changes: • Added the "When to Perform a PMA Analog Reset" table.
2018.10.08	Made the following changes: • Added the **RX Reset Duration** and **Use Separate TX/RX Reset Per Channel** parameters to the "Reset Parameters" table. • Added the "Reset Controller when **Use Separate TX/RX Reset Per Channel** is Turned OFF and **Enable Individual TX and RX Reset** is Turned OFF" figure. • Added the "PMA Analog Reset" section. • Added the "Reset Requirements when RS-FEC is Enabled" table and descriptions about necessary requirements when RS-FEC is enabled. • Changed the description of tx_reset in the "Reset Signals Required for E-Tile" table. • Updated the description for rx_transfer_ready and added the rx_pma_ready and rx_is_lockedtodata signals to the "Native PHY IP Ports With Manual Mode Enabled" table.

continued...

Document Version	Changes
	• Added a description about RS-FEC behavior in the "Manual Reset Mode" section. • Added the "Reset Controller Bypass" section and all subsections. • Added a signal to the "Reset Controller when **Use Separate TX/RX Reset Per Channel** is Turned OFF and **Enable Individual TX and RX Reset** is Turned OFF" figure. • Added the Enable TX/RX reset sequencing parameter to the "Reset Parameters" table.
2018.07.18	Made the following changes: • Updated the "RX Reset Deassertion Timing Waveform" sequence. • Moved AVMM to between first falling edge of rx_reset_req and rx_pma_ready rising edge in "RX PMA Reconfiguration with Reset Controller in Manual Mode Timing Waveform." • Moved AVMM to between first falling edge of tx_reset_req and tx_pma_ready rising edge in "TX PMA Reconfiguration with Reset Controller in Manual Mode Timing Waveform."
2018.05.15	Made the following changes: • Added the "Enable individual TX and RX reset" Reset Parameter. • Removed references to the sequencer. • Moved the "Resetting the E-Tile Transceiver" task and re-wrote it to encompass both analog and digital reset procedures. • Restructured "Automatic Reset Mode." • Added PMA ready signals to the "Manual Reset Mode" figure. • Added four Timing Waveforms: RX & TX Reset Assertion and RX & TX PMA Reconfiguration with Reset Controller in Manual Mode to the "Manual Reset Mode" section. • Removed PCS Gearbox and RS-FEC Direct from the "Reset Block Architecture" section because the PCS Gearbox is only for Interlaken and PMA direct with RS-FEC is in fractured mode. • Added a note to the "TX Reset Sequence in Automatic Mode After Power-Up" figure. • Added a note to the "RX Reset Sequence in Automatic Mode After Power-Up" figure. • Removed "PCS" from all sections.
2018.01.31	Initial release.

7. Dynamic Reconfiguration

Dynamic reconfiguration is the process of modifying transceiver channels to meet changing requirements during device operation. You can customize channels by triggering reconfiguration during device operation or following power-up.

Dynamic reconfiguration is available for E-tile Transceiver Native PHY.

Note: The Embedded Multi-die Interconnect Bridge (EMIB) can also be reconfigured in addition to channels using the reconfiguration interface.

Use the reconfiguration interface to dynamically change the transceiver channel settings and EMIB settings for the following applications.

- Fine tuning signal integrity by adjusting TX analog settings and RX adaptation settings
- Enabling or disabling transceiver channel blocks, such as the PRBS generator and verifier, and loopback modes
- Changing TX/RX settings for multi-data rate support protocols such as CPRI
- Enabling/disabling RS-FEC

The Native PHY IP cores provide the following features that allow dynamic reconfiguration:

- Reconfiguration interface
- Configuration files
- Multiple reconfiguration profiles
- Embedded reconfiguration streamer
- Native PHY Debug Master Endpoint (NPDME)
- Optional reconfiguration logic

Also see *Unsupported Features*.

The RS-FEC AVMM interface allows you to reconfigure the RS-FEC block and monitor status.

Related Information

Unsupported Features on page 152

7.1. Dynamically Reconfiguring Channel Blocks

Table 63. Dynamic Reconfiguration Feature Support

Reconfiguration	Features
Channel Reconfiguration	PMA Analog Features such as: • Attenuation Value (VOD) • Pre-emphasis • Enable RX adaptation
	Enable/disable RS-FEC. Read out RS-FEC statistics
	Reconfigure between NRZ and PAM4 when in PMA direct mode
	TX local clock dividers, reference clock
	RX local clock dividers, reference clock
	Clock output frequency

7.2. Dynamic Reconfiguration Maximum Data Rate Switch

If you use multiple transceiver channels in an E-tile, where at least one is configured in PAM4 high data rate (> 30 Gbps), you cannot change the data rate on other channels by greater than 20 Gbps (regardless of whether the channels are NRZ or PAM4) at a time in order to minimize the rate switch's effect on the PAM4 high data rate channel's link performance. You must step the data rate change, with a minimum step duration of 200 ms.

For example, if you have the following configuration:

- One PAM4 high data rate channel running at 56 Gbps (channel a)
- One NRZ channel running at 28 Gbps (channel b)

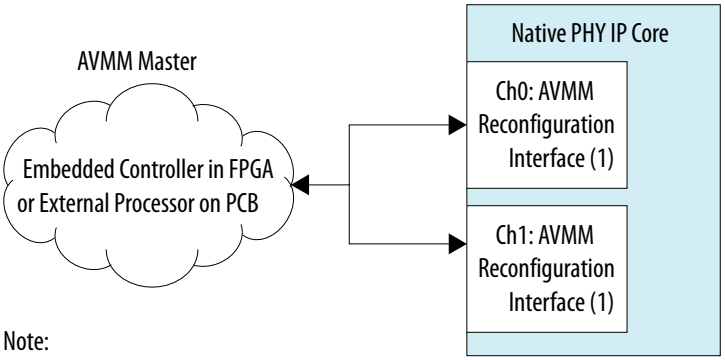
And you want to change the data rate on channel b from 28 Gbps to 5 Gbps, you must do it in two steps (first changing the data rate from 28 Gbps to 8 Gbps and then from 8 Gbps to 5 Gbps), with channel b running at 8 Gbps for a minimum of 200 ms. However, if you want to change the data rate on channel b from 28 Gbps to 25 Gbps, there is no restriction since the data rate change delta is less than 20 Gbps.

7.3. Interacting with the Dynamic Reconfiguration Interface

Each transceiver channel contains a reconfiguration interface shared with the PMA Interface (PMAIF), PMA and Embedded Multi-die Interconnect Bridge (EMIB). Additionally, there are six reconfiguration interfaces per E-tile allowing access to the six RS-FEC blocks.

The reconfiguration interface provides direct access to the programmable space of each channel. Communication with the channel reconfiguration interface requires an AVMM master. Because each channel has its own dedicated AVMM interface, you can dynamically reconfigure channels either concurrently or sequentially, depending on how the AVMM master is connected to the AVMM reconfiguration interface.

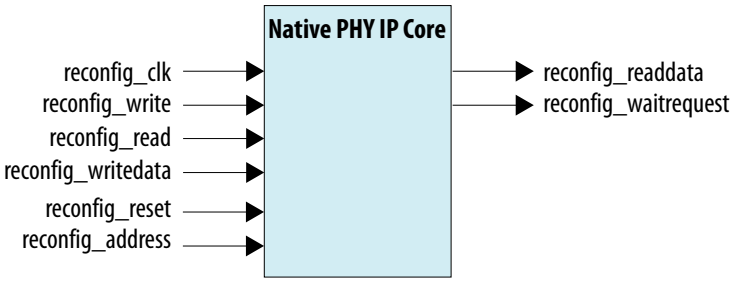
Figure 93. Reconfiguration Interface in Native PHY IP Cores



A Native PHY IP core instance can specify multiple channels. You can use a dedicated reconfiguration interface for each channel or share a single reconfiguration interface across multiple channels to perform dynamic reconfiguration.

AVMM masters interact with the reconfiguration interface by performing AVMM read and write operations to initiate a dynamic reconfiguration of specific transceiver parameters. The dynamic reconfiguration interfaces are compliant with AVMM specifications.

Figure 94. Reconfiguration Interface Ports with Shared Native PHY Reconfiguration Interface



7.4. Unsupported Features

Dynamic reconfiguration between the following modes is not supported by the Transceiver Native PHY IP core:

- PMA direct high data rate PAM4
- PMA Direct bonded mode
- Non RS-FEC to RS-FEC reconfiguration. You cannot create multiple dynamic reconfiguration profiles where RS-FEC is enabled in only certain profiles. You must enable or disable RS-FEC through the AVMM interface. See *RS-FEC Registers*.

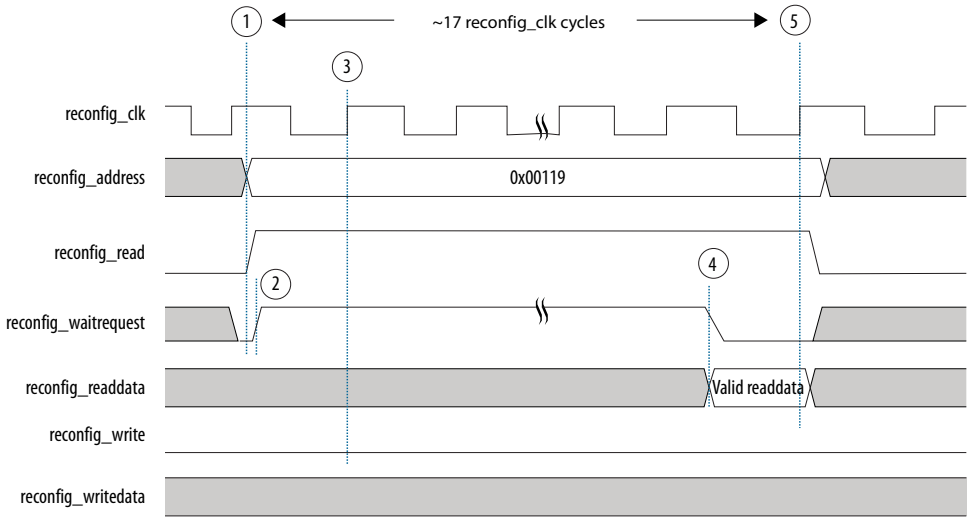
Related Information

RS-FEC Registers on page 247

7.5. Reading from the Dynamic Reconfiguration Interface

Reading from the reconfiguration interface of the Transceiver Native PHY IP core retrieves the current value at a specific address.

Figure 95. Reading from the Reconfiguration Interface

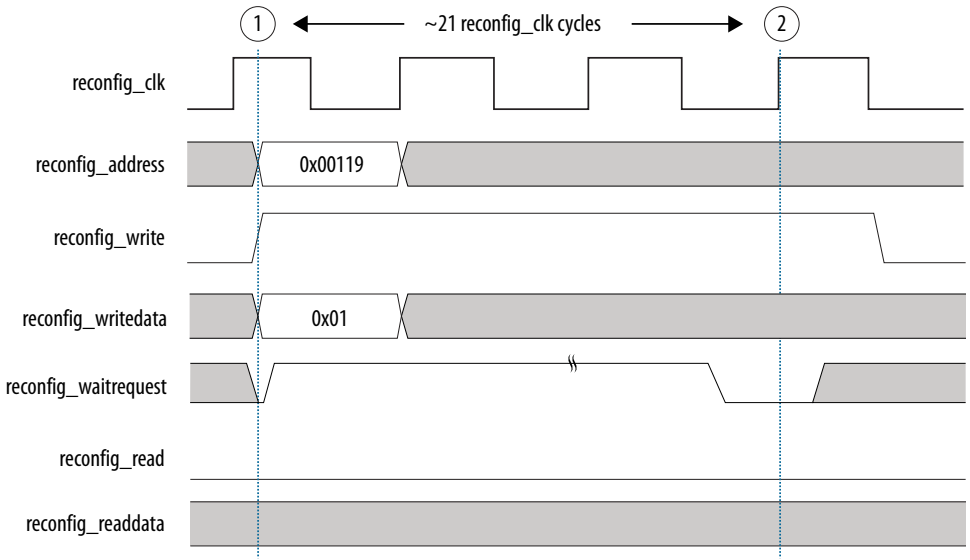


1. The master asserts reconfig_address and reconfig_read after the rising edge of reconfig_clk.
2. The slave asserts reconfig_waitrequest, stalling the transfer.
3. The master samples reconfig_waitrequest. Because reconfig_waitrequest is asserted, the cycle becomes a wait state and reconfig_address, reconfig_read, and reconfig_write remain constant.
4. The slave presents valid reconfig_readdata and deasserts reconfig_waitrequest.
5. The master samples reconfig_waitrequest and reconfig_readdata, completing the transfer.

7.6. Writing to the Dynamic Reconfiguration Interface

Writing to the reconfiguration interface of the Transceiver Native PHY IP core changes the data value at a specific address. All writes to the reconfiguration interface must be read-modify-writes, because two or more features may share the same reconfiguration address. You need to monitor the `reconfig_waitrequest` signal.

Figure 96. Writing to the Reconfiguration Interface



1. The master asserts the reconfig_address, reconfig_write, and reconfig_writedata signals.
2. The slave captures reconfig_writedata, ending the transfer.

7.7. Multiple Reconfiguration Profiles

You should enable multiple configurations or profiles in the same Native PHY IP for performing dynamic reconfiguration. This allows the IP **Parameter Editor** to create, store, and analyze the parameter settings for multiple configurations or profiles. The Native PHY can generate configuration files for all profiles in the SystemVerilog, MIF, or C header file formats. The files are located in the `<IP instance name>/altera_xcvr_native_s10_etile_181/synth/reconfig` subfolder of the IP instance with the configuration profile index added to the filename. For example, the configuration file for Profile 0 is stored as `<filename_CFG0.sv>`. The Intel Quartus Prime Timing Analyzer includes the necessary timing paths for all configurations based on initial and target profiles. You can also generate full reconfiguration files or reduced configuration files that contain only the attributes that differ between the multiple configured profiles. You can create up to eight reconfiguration profiles (Profile 0 to Profile 7) at a time for each Native PHY instance.

The configuration files generated by Native PHY IP also include PMA analog attributes.

You can use the multiple reconfiguration profiles feature without using the embedded reconfiguration streamer feature. When using the multiple reconfiguration profiles feature by itself, you must write the logic to reconfigure all entries that are different between the profiles while moving from one profile to another.

Note: You must ensure that none of the profiles in the Native PHY IP gives error messages, else the IP generation fails. The Native PHY IP core only validates the current active profile dynamically. For example, if you store a profile with error messages in the Native PHY IP and load another profile without any error messages, the error messages disappear in the IP. You are allowed to generate the IP, but the generation fails.

7.7.1. Reconfiguration Files

The E-tile Transceiver Native PHY IP core optionally allows you to save the parameters you specify for the IP instances as configuration files. The configuration file stores addresses and data values for that specific IP instance. The configuration files are generated during IP generation. They are located in the `<IP instance name>/altera_xcvr_native_s10_etile_181/synth/reconfig` subfolder of the IP instance. The configuration data is available in the following formats:

- SystemVerilog packages: <name>.sv
- C Header files: <name>.h
- Memory Initialization File (MIF): <name>.mif

Select one or more of the configuration file formats on the **Dynamic Reconfiguration** tab of the Transceiver Native PHY parameter editor to store the configuration data. The contents of the configuration files can be used to reconfigure from one transceiver configuration to another.

Reconfiguration files do not support going from non-RS-FEC mode to RS-FEC mode nor from RS-FEC mode to non-RS-FEC mode.

The generated MIF files do not contain the RS-FEC registers that need to be reconfigured.

Example 1. SystemVerilog Configuration File

```
package altera_xcvr_rcfg_10_reconfig_parameters_CFG0;

localparam ram_depth = 21;
function [34:0] get_ram_data;
  input integer index;
  automatic reg [0:20][34:0] ram_data = {
    35'h0380706, // [34:16]-DPRIO address=0x038; [15:8]-bit mask=0x07; [2:2]-
hssi_xcvr_cfg_rb_cont_cal=dcc_cont_cal_en(1'h1); [1:1]-
hssi_xcvr_cfg_rb_dcc_en=dcc_mast_en(1'h1); [0:0]-
hssi_xcvr_cfg_rb_dcc_byp=dcc_byp_dis(1'h0);
    35'h03C0202, // [34:16]-DPRIO address=0x03C; [15:8]-bit mask=0x02; [1:1]-
hssi_xcvr_cfg_dcc_csr_en_fsm=dcc_en_fsm(1'h1);
    35'h0A4FF40, // [34:16]-DPRIO address=0x0A4; [15:8]-bit mask=0xFF; [7:0]-
hssi_xcvr_int_seq3_tx_refclk_ratio=64(8'h40);
    35'h0A8FF40, // [34:16]-DPRIO address=0x0A8; [15:8]-bit mask=0xFF; [7:0]-
hssi_xcvr_int_seq4_rx_refclk_ratio=64(8'h40);
    35'h21A8080, // [34:16]-DPRIO address=0x21A; [15:8]-bit mask=0x80; [7:7]-
hssi_adapt_rx_word_mark=wm_en(1'h1);
    35'h2310400, // [34:16]-DPRIO address=0x231; [15:8]-bit mask=0x04; [2:2]-
hssi_aibcr_tx_aib_dllstr_align_dy_ctlsel=aib_dllstr_al
```

Note: DPRIO refers to AVMM addresses.

The SystemVerilog configuration files contain two parts. The first part consists of a data array of 35-bit hexadecimal values. The second part consists of parameter values. For the data array, each 35-bit hexadecimal value is associated with a comment that describes the various bit positions.

Table 64. Mapping of SystemVerilog Configuration File Line

Bit Position	Description
[34:16]	The AVMM address.
[15:8]	The AVMM bit mask. The bit mask exposes the bits that are configured in either the Transceiver Native PHY IP cores.
[7:0]	Feature bit values.

For example, a value of 35'h002310400 represents an address of 0x00231 and a bit mask of 0x04. There is a feature located on bit 2 named `hssi_aibcr_tx_aib_dllstr_align_dy_ctlsel`, and it is set to `aib_dllstr_al`, which has the value of 0. The MIF file and C header file are set up similarly to the SystemVerilog package file. Multiple transceiver features may reside at the same address. Also, a single transceiver feature may span across multiple addresses.

You can generate multiple configurations (up to eight) of the transceiver Native PHY IP core. You can select any configuration as the default power-up configuration.

7.7.2. Embedded Reconfiguration Streamer

You can optionally enable the embedded reconfiguration streamer in the Native PHY IP cores to automate the reconfiguration operation. The embedded reconfiguration streamer is a feature block that can perform Avalon-MM transactions to access channel configuration registers in the transceiver. When you enable the embedded streamer, the Native PHY IP cores embed HDL code for reconfiguration profile storage and reconfiguration control logic in the IP files.

If the new profile requires changing PMA attributes that can only be performed when the PMA is disabled, you need to do the following:

1. Assert digital reset.
2. Disable the PMA using PMA attribute code 0x0001.
3. Write to AVMM register 0x40140 with the following bit pattern:
   - Bits [2:0] equal to the new profile
   - Bit [7] equal to 1 to launch the reconfiguration streamer

   The reconfiguration streaming automatically requests PMA recalibration.
4. Continuously read register 0x40141[0]. It asserts high while loading the new profile and goes low after the new profile has finished loading. The PMA is now enabled and calibrated.
5. Deassert digital reset.

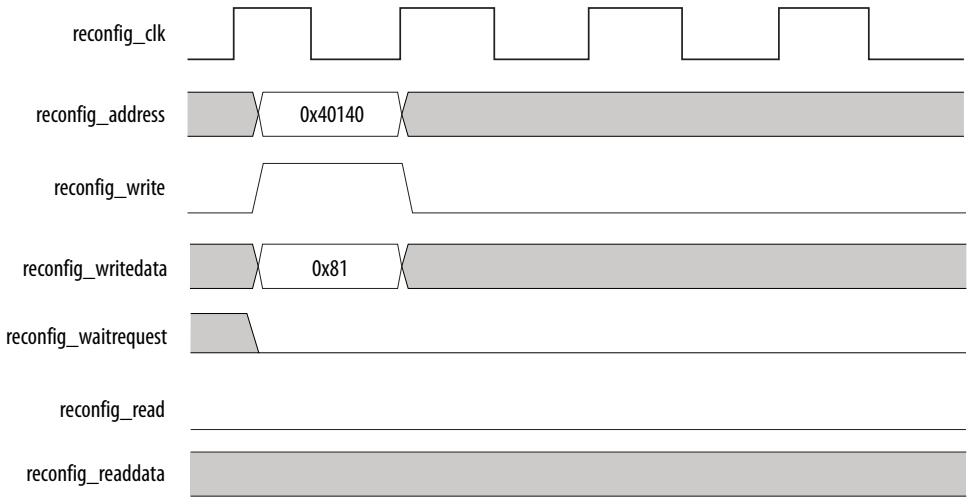
If the new profile does not require the PMA to be disabled, you need to do the following:

1. Write to AVMM register 0x40140 with the following bit pattern:
   - Bits [2:0] equal to the new profile
   - Bit [7] equal to 1 to launch the reconfiguration streamer
2. Continue to read register 0x40141[0] until it becomes 0 to indicate that the reconfiguration streamer is finished.

7.7.2.1. Register 0x40140

The AVMM `waitrequest` signal does not assert when writing to the register to start the MIF streaming because the register is located in the FPGA core.

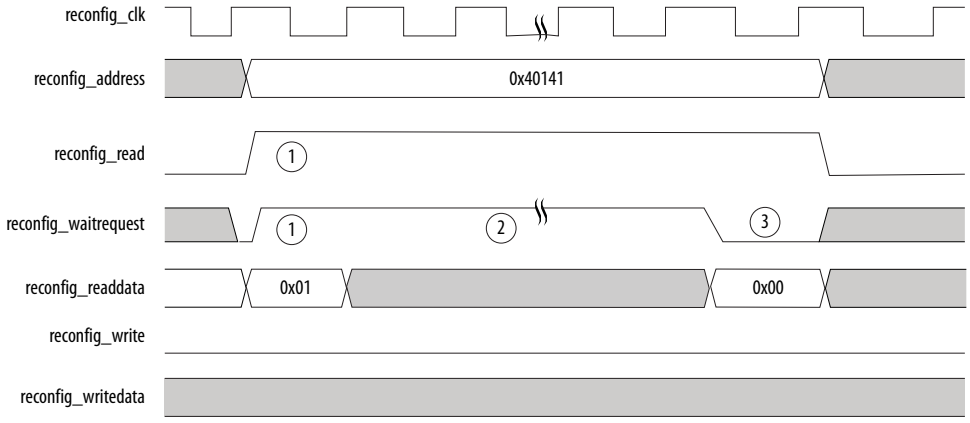
Figure 97. Loading MIF Profile 1 by Writing to Register 0x40140



7.7.2.2. Register 0x40141

Bit 0 pulses high to indicate that MIF streaming is in progress. However, you only have ~400 AVMM clock cycles after writing to register 0x40140 to read bit 0 as high (1 in the below figure) before the MIF streamer asserts the AVMM `waitrequest` and locks the AVMM bus (2 in the below figure). After it releases the bus, bit 0 becomes 0, and the new profile is loaded (3 in the below figure).

Figure 98. MIF Streaming High Level Diagram Showing AVMM Bus Behavior and Register 0x40141 Status



1. Assert reconfig_read, and read back 0x01 on reconfig_readdata. Reconfig_waitrequest asserts on read.
2. MIF streamer asserts the reconfig_waitrequest to load the new Native PHY configuration. The AVMM bus is not available.
3. MIF streamer releases reconfig_waitrequest and clears 0x40141[0]. The AVMM bus is available.

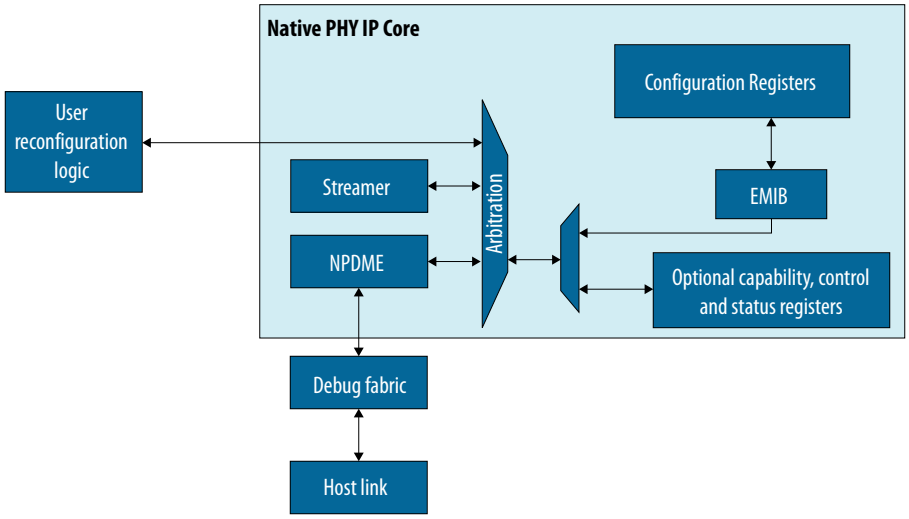
Refer to *Reading from the Dynamic Reconfiguration Interface* for details.

Related Information

Reading from the Dynamic Reconfiguration Interface on page 153

7.8. Arbitration

Figure 99. Arbitration



The arbitration logic allows multiple masters to control the AVMM bus. The following feature blocks can access the programmable registers:

- Embedded reconfiguration streamer
- NPDME
- User reconfiguration logic connected to the reconfiguration interface

These feature blocks arbitrate for control over the programmable space of each transceiver channel. Each of these feature blocks can request access to the programmable registers of a channel by performing a read or write operation to that channel. For any of these feature blocks to be used, you must first have control over the internal configuration bus.

The embedded reconfiguration streamer has the highest priority, followed by the reconfiguration interface, followed by the NPDME. When two feature blocks are trying to access the same transceiver channel on the same clock cycle, the feature block with the highest priority is given access. The only exception is when a lower priority feature block is in the middle of a read/write operation and a higher priority feature block tries to access the same channel. In this case, the higher-priority feature block must wait until the lower-priority feature block finishes the read/write operation.

Note: When you enable NPDME in your design, you must either:

- Connect an AVMM master to the reconfiguration interface.
- Connect the `reconfig_clock` and `reconfig_reset` ports, and ground the `reconfig_write`, `reconfig_read`, `reconfig_address`, and `reconfig_writedata` ports of the reconfiguration interface. If you do not connect the reconfiguration interface signals appropriately, the NPDME does not function properly.

7.9. Recommendations for PMA Dynamic Reconfiguration

Some PMA features that can be dynamically reconfigured, like the reference clock source, the TX and RX data rate require the PMA and digital blocks to be in the reset state. Intel recommends that you:

- Hold the channel transmitter in digital reset and assert PMA attribute codes to disable the PMA TX during reconfiguration.
- Hold the channel receiver in digital reset and assert PMA attribute codes to disable the CDR during reconfiguration.

7.10. Steps to Perform Dynamic Reconfiguration

Refer to *PMA Attribute Codes* for information about which PMA attributes codes require the PMA to be reset.

Figure 100. Dynamic Reconfiguration with Native PHY in Automatic Reset Mode

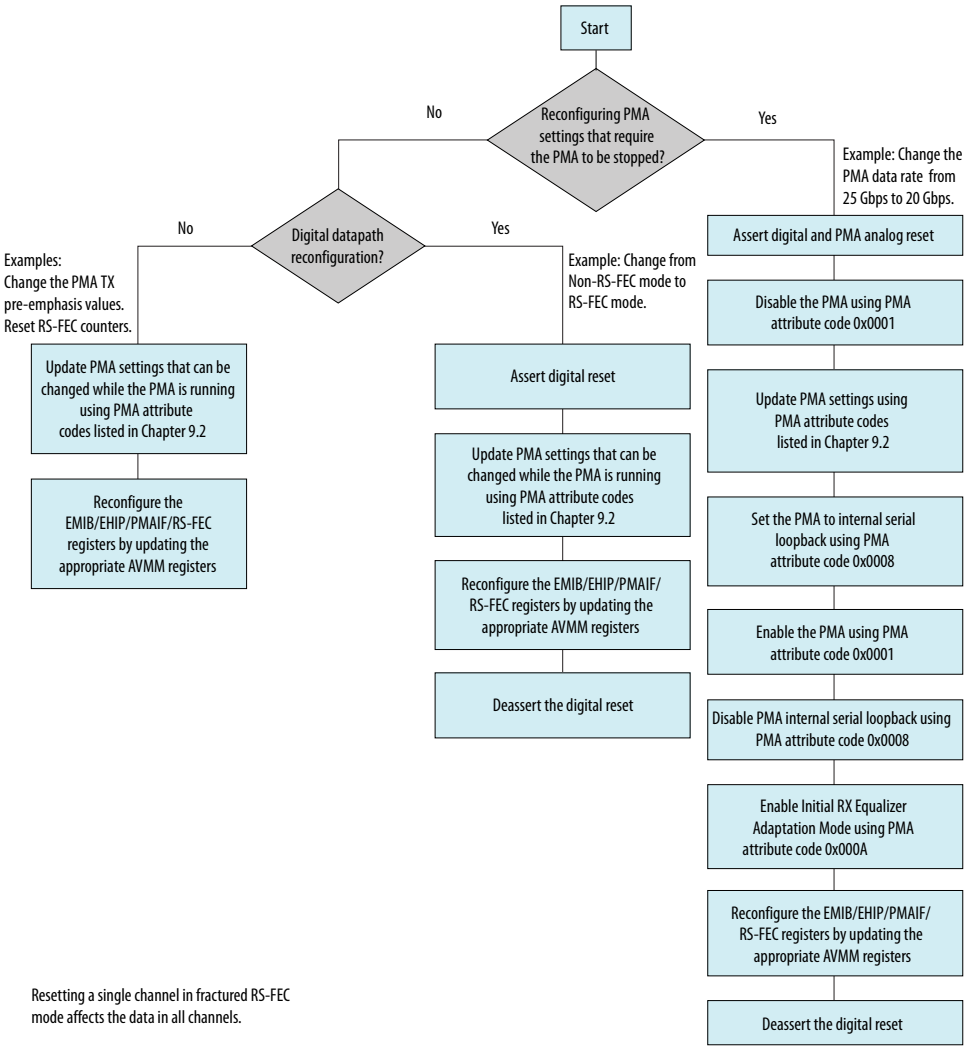
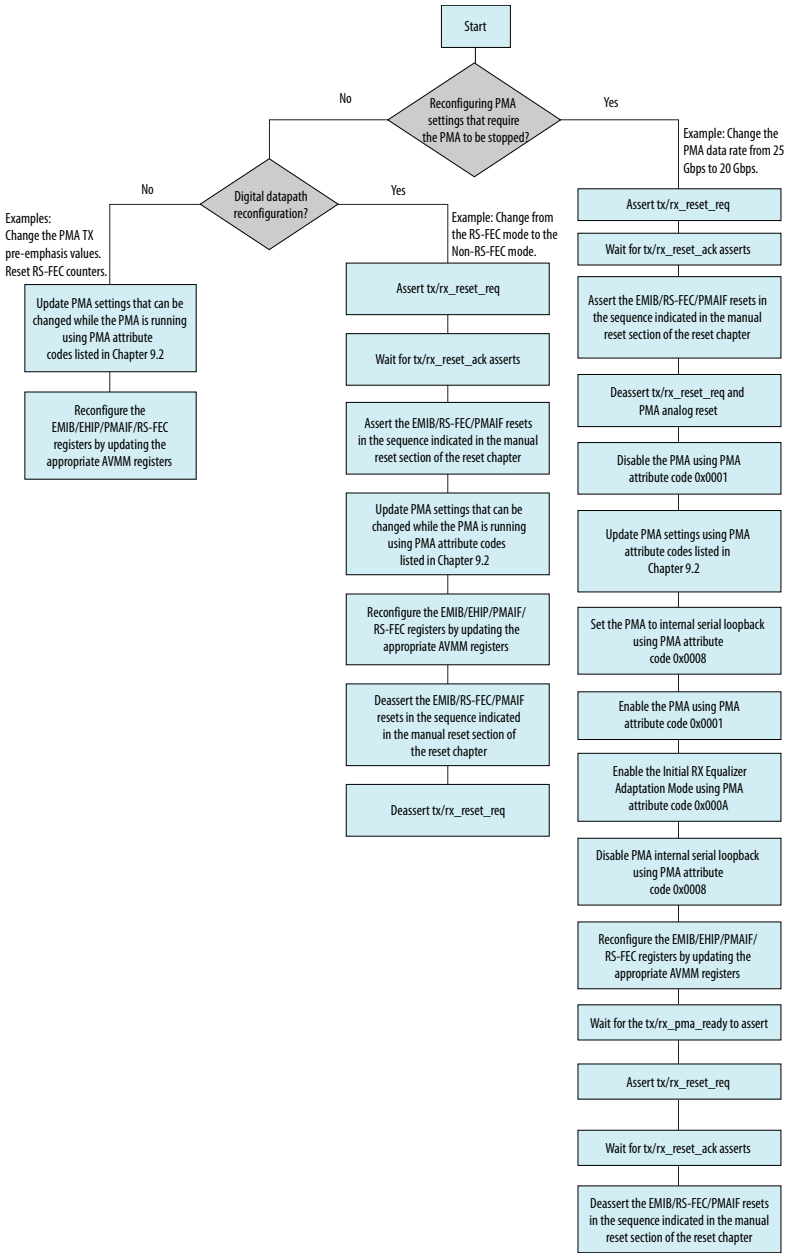


Figure 101. Dynamic Reconfiguration with Native PHY in Manual Reset Mode (required for fractured RS-FEC mode)



Related Information

PMA Attribute Codes on page 216

7.11. PMA Attribute Details

Use the following steps to update the PMA analog settings by changing the PMA attributes in the PMA AVMM interface. Refer to *PMA Attribute Codes* for a list of the PMA attribute features.

1. To modify PMA settings using PMA attribute codes, write the appropriate PMA attribute code and data to the PMA AVMM reconfiguration interface registers 0x87 to 0x84.
2. Issue a PMA attribute code request by setting 0x90[0] to 1.
3. Verify that the PMA attribute code is sent to the PMA by verifying that 0x8A[7] is asserted.
4. Verify that 0x8B[0] deasserts to indicate that the PMA attribute code transaction completed.
5. Read 0x89 to 0x88 if the PMA attribute code is expected to return data.
6. Write 0x8A[7] to 1 to clear the 0x8A[7] value.
7. Repeat steps 1 to 6 for additional PMA attribute codes.

Note: Address 0x8A[7] is asserted upon power up. You must clear the bit by writing 0x8A[7] to 1 before writing any attributes to the PMA.

Related Information

- PMA Register Map on page 211
- PMA Attribute Codes on page 216

7.12. Dynamic Reconfiguration Flow for Special Cases

Dynamic reconfiguration can be performed on logical operations such as switching between multiple reference clocks. In these cases, configuration files alone cannot be used. Configuration files are generated during IP generation and do not contain information on the placement of reference clocks.

To perform dynamic reconfiguration on logical operations, you must use lookup registers that contain information about logical index to physical index mapping. Lookup registers are read-only registers. Use these lookup registers to perform a read-modify-write to the selection MUXes to switch between reference clocks.

To perform dynamic reconfiguration using reconfiguration flow for special cases:

1. Read from the desired lookup register.
2. Perform logical encoding.
3. Perform read-modify-write to the required feature address with the desired/ encoded value.

7.12.1. Switching Reference Clocks

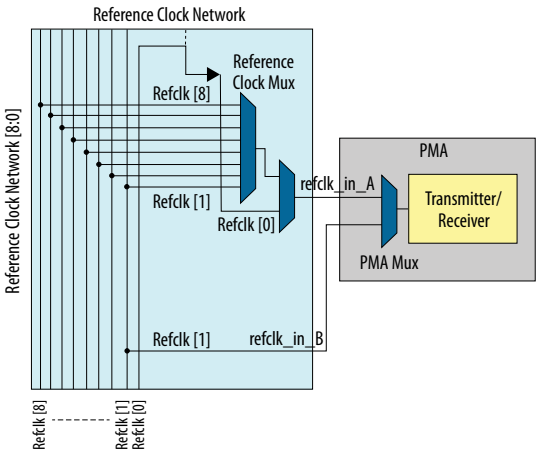
You can dynamically switch the input clock source.

You can use the reconfiguration interface on the channel instance to specify which reference clock source drives the transmitter, the receiver, or both. The channel supports clocking up to five different reference clock sources of the nine available clocking sources from the reference clock network.

Before initiating a reference clock switch, ensure that your Native PHY instance defines more than one reference clock source. Specify the **Number of reference clocks inputs per channel** parameter.

The number of exposed `refclk` ports varies according to the number of reference clocks you specify. Use the reconfiguration interface to look up the mux settings for the different `refclk#` and write the look up value into the channel. Refer to *PMA AVMM Registers* for look up values.

Figure 102. Reference Clock Network



The reference clock network uses **Reference Clock Mux** by default which means that:

- `refclk_in_A` is the default reference clock at power-up.
- `refclk_in_A` is the default reference clock after a PMA reset.

The most frequent reference clock dynamic reconfiguration use cases are:

- Switching between any two `refclk[0, 1, 2, 3, 4, 5, 6, 7]` reference clocks or changing the reference clock frequency on `refclk[0, 2, 3, 4, 5, 6, 7, 8]`.
- Switching from `refclk[0, 1, 2, 3, 4, 5, 6, 7]` to `refclk[8]`.
- Changing the `refclk[1]` reference clock frequency.

For correct dynamic reconfiguration on reference clocks, find the procedure applicable to your use case in the sections below. Any change in the input reference clock frequency must follow these procedures and must not directly change the reference clock frequency.(30), (31)

(30) The reference clock frequency range is from 125 MHz to 700 MHz as per the *Device Data Sheet*.

Requirements for all procedures:

- A minimum of two clocks bonded out, and `refclk[1]` must always be bonded out.
- At least one clock must be stable at a given time.
- Switch **PMA Mux** to a stable clock before switching the reference clock mux.

Refer to *PMA Attribute Details* for how to write a PMA attribute code.

Related Information

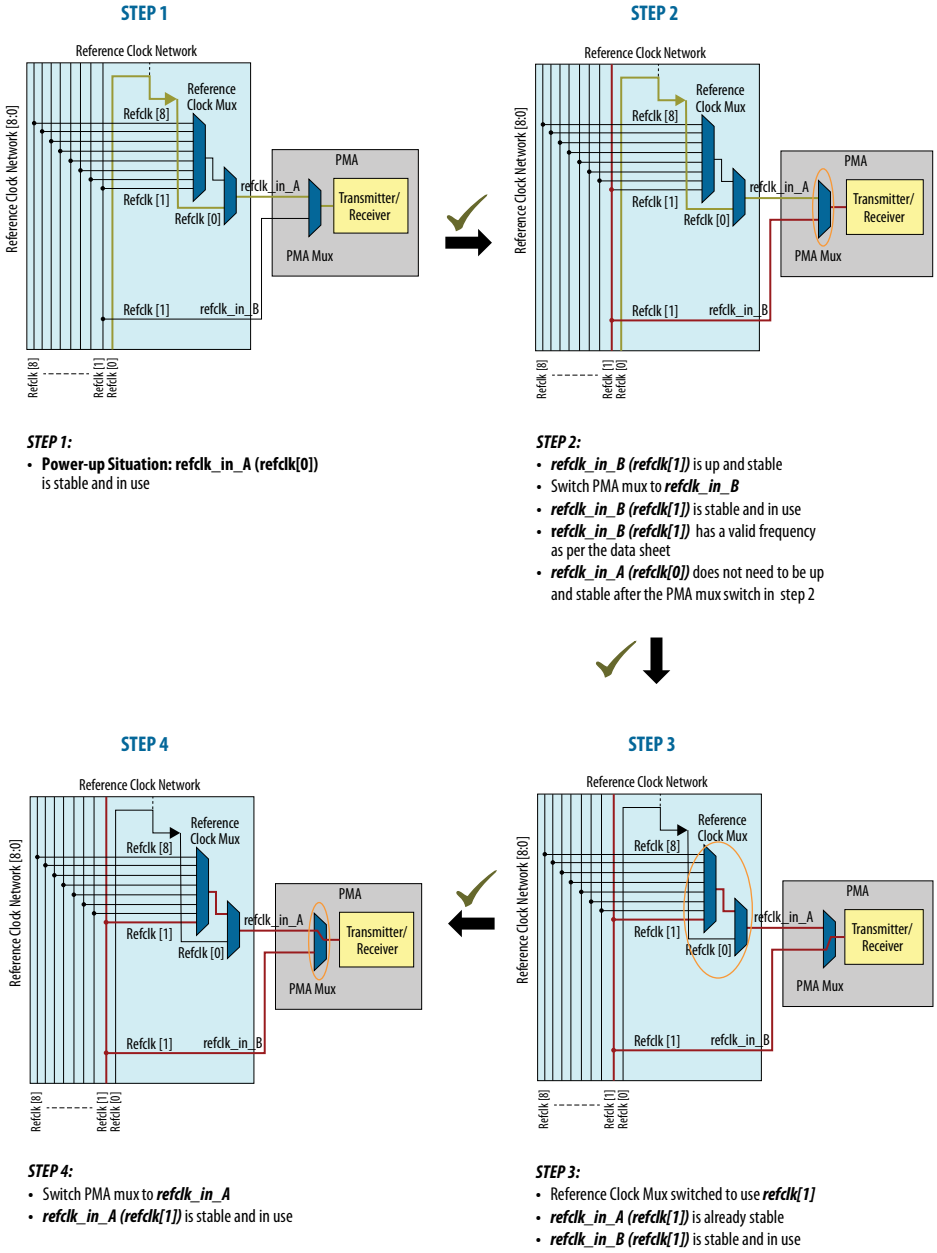
- PMA Attribute Details on page 161
- PMA AVMM Registers on page 213
- Intel Stratix 10 Device Data Sheet
- Intel Agilex Device Data Sheet

(31) PPM variation by protocols is not considered a frequency change.

7.12.1.1. Switching Between Any Two refclk[0, 1, 2, 3, 4, 5, 6, 7] Reference Clocks or Changing the Reference Clock Frequency on refclk[0, 2, 3, 4, 5, 6, 7, 8]

1. Device bring-up defaults the reference clock to refclk_in_A.
2. Bring up refclk_in_B (refclk[1]), and switch **PMA Mux** to refclk_in_B (**PMA Mux** is now working on refclk_in_B (refclk[1]).
   - Make sure the refclk_in_B (refclk[1]) frequency is valid per the data sheet.
   - Write PMA attribute 0x30 = 0x3 to switch the **PMA Mux**.
3. Switch **Reference Clock Mux** (9:1 mux) based on your requirement, or change the reference clock frequency (**PMA Mux** is still working on refclk_in_B (refclk[1]). For changing the reference clock frequency:
   a. If the old and new frequencies are both in the same range of 125-500 MHz or both in the same range 500-700 MHz, you only need to switch one reference clock from the old value to the new value.
   b. If, however, you are switching between 0-500 MHz and 500-700 MHz, you must use two reference clocks to represent the old and new clock frequencies, respectively.
4. Bring up refclk_in_A, and switch **PMA Mux** to refclk_in_A.
   - Write PMA Attribute 0x30 = 0x0 to switch the **PMA Mux**.
5. Perform a *PMA Analog Reset*.

Example 2. Reference Clock Switch from `refclk[0]` to `refclk[1]`



STEP 1:

- **Power-up Situation: refclk_in_A (refclk[0])** is stable and in use

STEP 2:

- ***refclk_in_B (refclk[1])*** is up and stable
- Switch PMA mux to ***refclk_in_B***
- ***refclk_in_B (refclk[1])*** is stable and in use
- ***refclk_in_B (refclk[1])*** has a valid frequency as per the data sheet
- ***refclk_in_A (refclk[0])*** does not need to be up and stable after the PMA mux switch in step 2

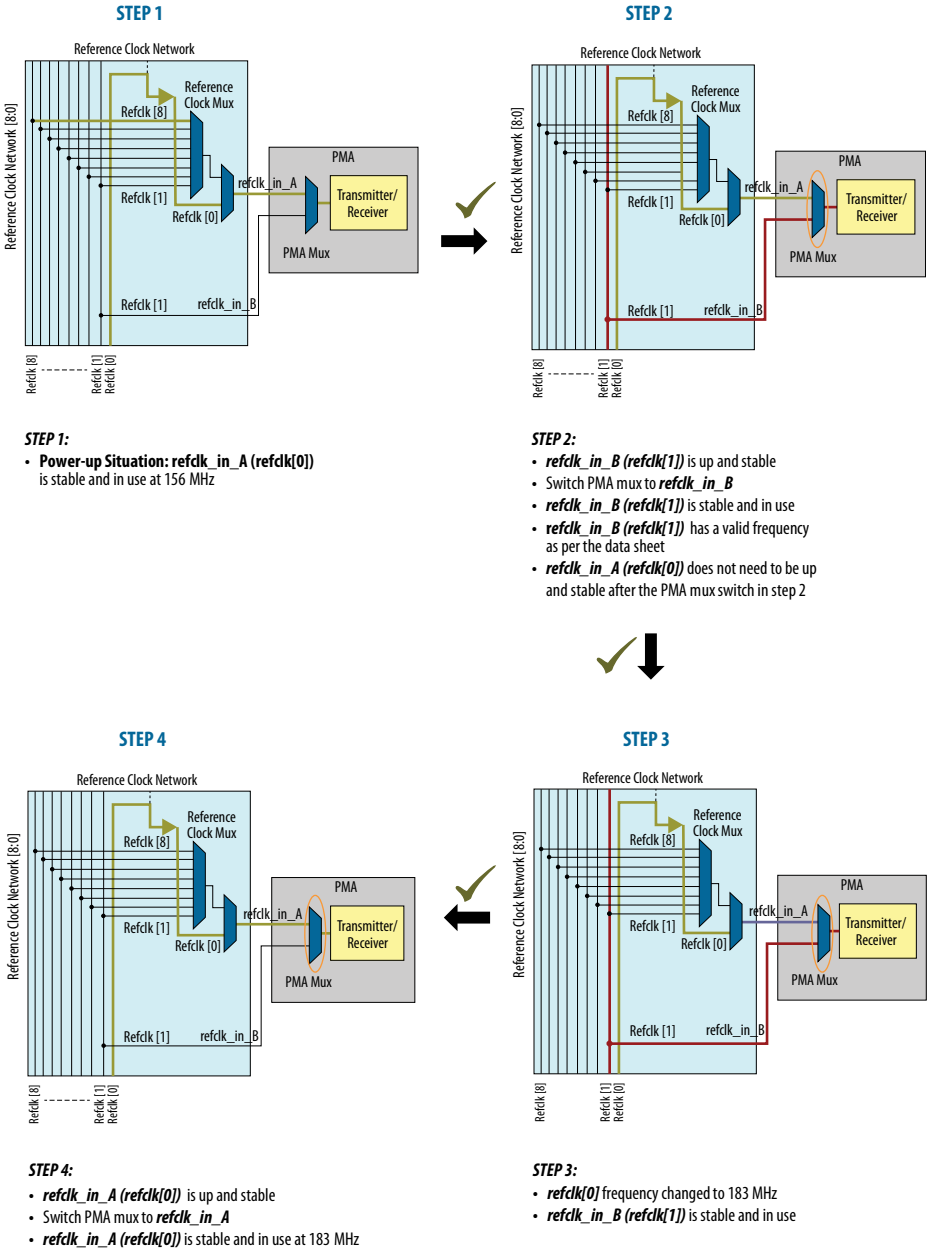
STEP 3:

- Reference Clock Mux switched to use ***refclk[1]***
- ***refclk_in_A (refclk[1])*** is already stable
- ***refclk_in_B (refclk[1])*** is stable and in use

STEP 4:

- Switch PMA mux to ***refclk_in_A***
- ***refclk_in_A (refclk[1])*** is stable and in use

Example 3. refclk[0] Frequency Change from 156 MHz to 183 MHz



STEP 1:

- **Power-up Situation: refclk_in_A (refclk[0])** is stable and in use at 156 MHz

STEP 2:

- ***refclk_in_B (refclk[1])*** is up and stable
- Switch PMA mux to ***refclk_in_B***
- ***refclk_in_B (refclk[1])*** is stable and in use
- ***refclk_in_B (refclk[1])*** has a valid frequency as per the data sheet
- ***refclk_in_A (refclk[0])*** does not need to be up and stable after the PMA mux switch in step 2

STEP 4:

- ***refclk_in_A (refclk[0])*** is up and stable
- Switch PMA mux to ***refclk_in_A***
- ***refclk_in_A (refclk[0])*** is stable and in use at 183 MHz

STEP 3:

- ***refclk[0]*** frequency changed to 183 MHz
- ***refclk_in_B (refclk[1])*** is stable and in use

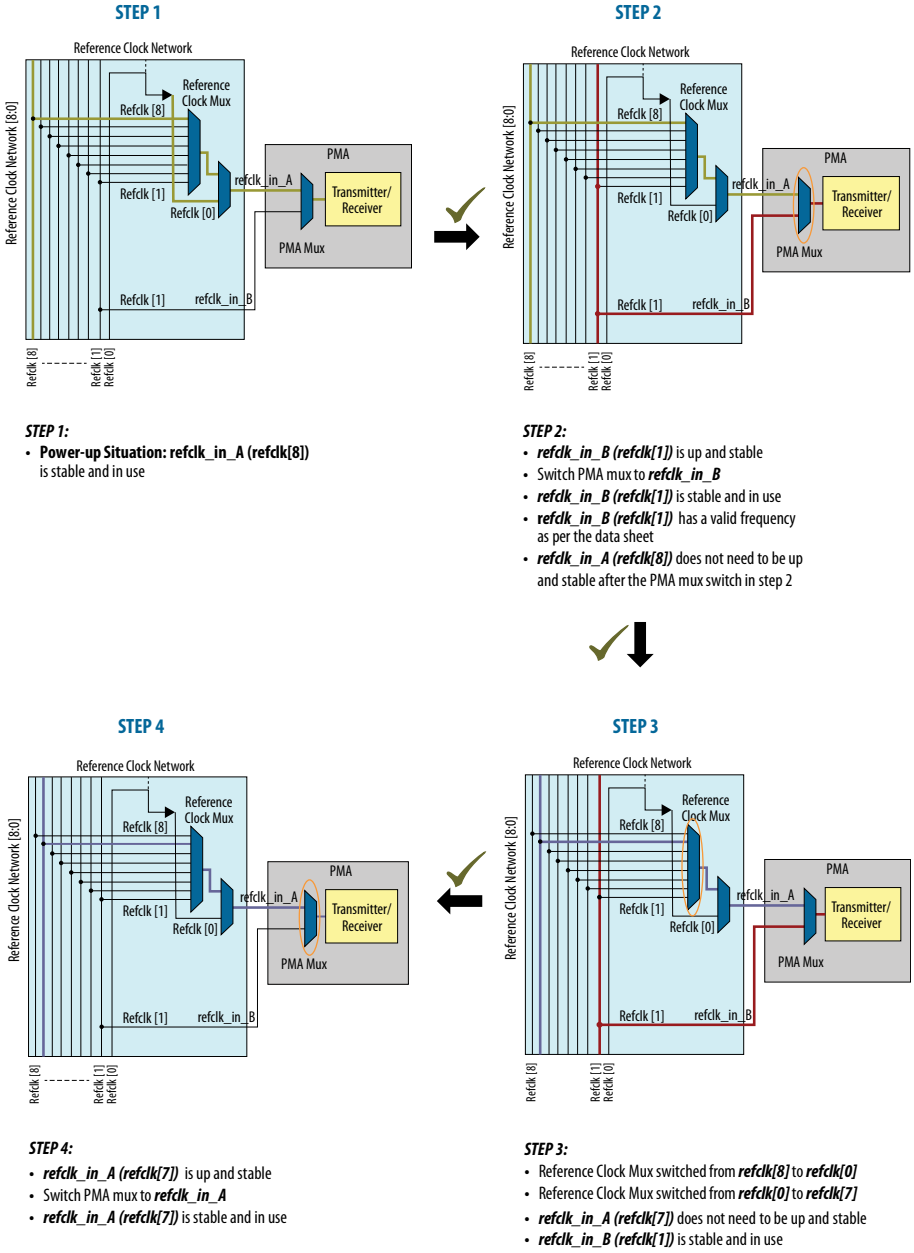
Related Information

PMA Analog Reset on page 132

7.12.1.2. Switching from/to `refclk[0, 1, 2, 3, 4, 5, 6, 7]` to/from `refclk[8]`

1. Device bring-up defaults the reference clock to `refclk_in_A`.
2. Bring up `refclk_in_B` (`refclk[1]`), and switch **PMA Mux** to `refclk_in_B` (**PMA Mux** is now working on `refclk_in_B` (`refclk[1]`).
   - Make sure the `refclk_in_B` (`refclk[1]`) frequency is valid per the data sheet.
   - Write PMA attribute 0x30 = 0x3 to switch the **PMA Mux**.
3. Switch **Reference Clock Mux** (9:1 mux) to `refclk[0]`; then switch to the target reference clock (**PMA Mux** is still working on `refclk_in_B` (`refclk[1]`).
   - `refclk[0]` does not have to be bonded out on the board.
4. Bring up `refclk_in_A`, and switch **PMA Mux** to `refclk_in_A`.
   - Write PMA Attribute 0x30 = 0x0 to switch the **PMA Mux**.
5. Perform a *PMA Analog Reset*.

Example 4. Switching from/to refclk[0, 1, 2, 3, 4, 5, 6, 7] to/from refclk[8]



STEP 1:

- **Power-up Situation: refclk_in_A (refclk[8])** is stable and in use

STEP 2:

- ***refclk_in_B (refclk[1])*** is up and stable
- Switch PMA mux to ***refclk_in_B***
- ***refclk_in_B (refclk[1])*** is stable and in use
- ***refclk_in_B (refclk[1])*** has a valid frequency as per the data sheet
- ***refclk_in_A (refclk[8])*** does not need to be up and stable after the PMA mux switch in step 2

STEP 4:

- ***refclk_in_A (refclk[7])*** is up and stable
- Switch PMA mux to ***refclk_in_A***
- ***refclk_in_A (refclk[7])*** is stable and in use

STEP 3:

- Reference Clock Mux switched from ***refclk[8]*** to ***refclk[0]***
- Reference Clock Mux switched from ***refclk[0]*** to ***refclk[7]***
- ***refclk_in_A (refclk[7])*** does not need to be up and stable
- ***refclk_in_B (refclk[1])*** is stable and in use

Related Information

PMA Analog Reset on page 132

7.12.1.3. Changing the refclk[1] Reference Clock Frequency

1. Device bring-up defaults the reference clock to refclk_in_A (refclk[1] in this use case).
2. Switch **PMA Mux** to refclk_in_B. Maintain refclk_in_B (refclk[1]) to be stable.
   - Write PMA attribute 0x30 = 0x3 to switch the **PMA Mux**.
3. Switch **Reference Clock Mux** (9:1 mux) from refclk[1] to a stable refclk[0, 2, 3, 4, 5, 6, 7].
4. Bring up refclk_in_A, and switch **PMA Mux** to refclk_in_A.
   - Write PMA Attribute 0x30 = 0x0 to switch the **PMA Mux**.
5. Change the frequency on refclk[1].
6. Bring up refclk_in_B, and switch **PMA Mux** to refclk_in_B.
   - Write PMA attribute 0x30 = 0x3 to switch the **PMA Mux**.
7. Switch **Reference Clock Mux** (9:1 mux) to refclk[1].
8. Because refclk_in_A is already stable, switch **PMA Mux** to refclk_in_A.
   - Write PMA Attribute 0x30 = 0x0 to switch the **PMA Mux**.
9. Perform a *PMA Analog Reset*.

Related Information

PMA Analog Reset on page 132

7.12.2. Reconfiguring PMA Settings

The following steps allow you to reconfigure PMA settings as needed.

1. Set 0x91 to 0x01 if you want to:
   - Go to the initial PMA configuration (when the embedded reconfiguration streamer is not used)
   - Go to the last selected profile (when the embedded reconfiguration streamer is used)

   0x91[0] automatically clears once the PMA is loaded with the correct settings.
2. If you want to change the PMA to a new configuration, you must send the following attributes in the following order:
   a. Set the TX data rate to reference clock ratio by sending attribute code 0x0005.
   b. Set the RX data rate to reference clock ratio by sending attribute code 0x0006.
   c. Set the PMA's serializer/deserialzer ratios and NRZ/PAM4 by sending attribute code 0x0014.
   d. Set the TX equalization by sending attribute code 0x0015.
   e. Request PMA calibration when it is enabled by sending attribute code 0x0011.
   f. Bring the PMA out of reset by sending attribute code 0x0001.
3. Bring the Native PHY IP core out of digital reset.

Note: 0x8A[7] is 1 after either using the embedded reconfiguration stream to change to a new profile, or using register 0x91[0] to restore the previous profile. You must clear register 0x8A[7] by writing 0x8A[7] to 1 before sending any attributes to the PMA.

7.13. Ports and Parameters

The reconfiguration interface is integrated in the Native PHY instance. Instantiate the Native PHY IP cores in the **IP Parameter Editor** by clicking **Tools ➤ IP Catalog**. You can define parameters for IP cores by using the IP-core-specific **Parameter Editor**. To expose the reconfiguration interface ports, select the **Enable dynamic reconfiguration** option when parameterizing the IP core. You can share the reconfiguration interface among all the channels by turning on **Share reconfiguration interface** when parameterizing the IP core. When this option is enabled, the IP core presents a single reconfiguration interface for the dynamic reconfiguration of all channels. Address bits [18:0] provide the register address in the reconfiguration space of the selected channel. The remaining address bits of the reconfiguration address specify the selected logical channel. For example, if there are four channels in the Native PHY IP instance, `reconfig_address[18:0]` specifies the address and `reconfig_address[20:19]` are binary encoded to specify the four channels. For example, 2'b01 in `reconfig_address[20:19]` specifies logical channel 1.

The following figure shows the signals available when the Native PHY IP core is configured for four channels and the **Share reconfiguration interface** option is enabled.

Figure 103. Reconfiguration Interface Ports with Shared Native PHY Reconfiguration Interface

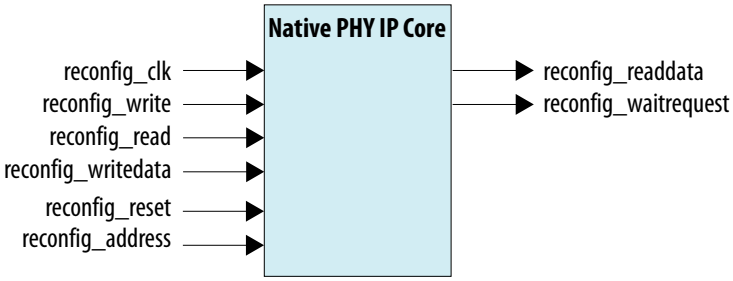


Table 65. Reconfiguration Interface Ports with Shared Native PHY Reconfiguration Interface

The reconfiguration interface ports when **Share reconfiguration interface** is enabled. <*N*> represents the number of channels.

Port Name	Direction	Clock Domain	Description
`reconfig_clk`	Input	N/A	The clock frequency is 100 to 162 MHz.
`reconfig_reset`	Input	`reconfig_clk`	Resets the AVMM interface. Asynchronous assertion and synchronous deassertion.
`reconfig_write`	Input	`reconfig_clk`	Write enable signal. Signal is active high.

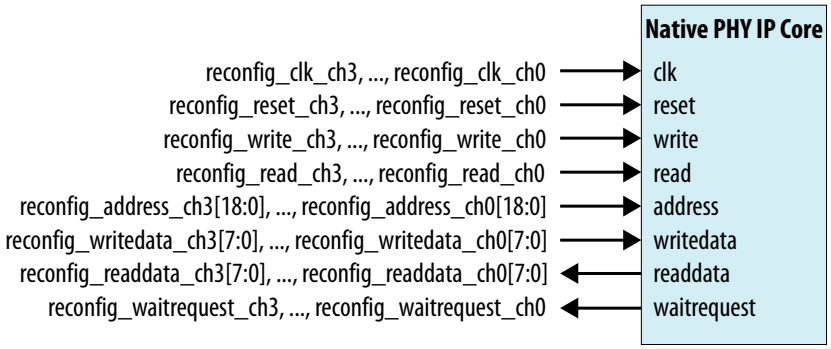
continued...

Port Name	Direction	Clock Domain	Description
`reconfig_read`	Input	`reconfig_clk`	Read enable signal. Signal is active high.
`reconfig_address[log2<N>+18:0]`	Input	`reconfig_clk`	Address bus. The lower 19 bits specify address, and the upper bits specify the channel.
`reconfig_writedata[7:0]`	Input	`reconfig_clk`	An 8-bit data write bus. Data to be written into the address indicated by `reconfig_address`.
`reconfig_readdata[7:0]`	Output	`reconfig_clk`	An 8-bit data read bus. Valid data is placed on this bus after a read operation. Signal is valid after `reconfig_waitrequest` goes high and then low.
`reconfig_waitrequest`	Output	`reconfig_clk`	A one-bit signal that indicates that the AVMM interface is busy. Keep the AVMM command asserted until the interface is ready to proceed with the read/ write transfer.

When **Share reconfiguration interface** is disabled and **Provide separate interface for each channel** is enabled, the Native PHY IP core provides an independent reconfiguration interface for each channel. For example, when a reconfiguration interface is not shared for a four-channel Native PHY IP instance, `reconfig_address_ch0[18:0]` corresponds to the reconfiguration address bus of logical channel 0, `reconfig_address_ch1[18:0]` correspond to the reconfiguration address bus of logical channel 1, `reconfig_address_ch2[18:0]` corresponds to the reconfiguration address bus of logical channel 2, and `reconfig_address_ch3[18:0]` correspond to the reconfiguration address bus of logical channel 3.

The following figure shows the signals available when the Native PHY is configured for four channels and the **Share reconfiguration interface** option is not enabled and **Provide separate interface for each channel** is enabled.

Figure 104. Signals Available with Independent Native PHY Reconfiguration Interfaces



Note: The RS-FEC reconfiguration interface is not separated for each channel.

Table 66. Reconfiguration Interface Ports with Independent Native PHY Reconfiguration Interface

The reconfiguration interface ports when **Share reconfiguration interface** is disabled. <*N*> represents the number of channels.

Port Name	Direction	Clock Domain	Description
reconfig_clk_ch<N-1>, ...,reconfig_clk_ch0	Input	N/A	The clock frequency is 100-162 MHz.
reconfig_reset_ch<N-1>, ...,reconfig_reset_ch0	Input	reconfig_clk_ch#	Resets the AVMM interface. Asynchronous assertion and synchronous deassertion.
reconfig_write_ch<N-1>, ...,reconfig_write_ch0	Input	reconfig_clk_ch#	Write enable signal. Signal is active high.
reconfig_read_ch<N-1>, ...,reconfig_read_ch0	Input	reconfig_clk_ch#	Read enable signal. Signal is active high.
reconfig_address_ch<N-1>[18:0], ...,reconfig_address_ch0[18:0]	Input	reconfig_clk_ch#	An 20-bit address bus for each channel.
reconfig_writedata_ch<N-1>[7:0], ...,reconfig_writedata_ch0[7:0]	Input	reconfig_clk_ch#	A 8-bit data write bus for each channel. Data to be written into the address indicated by reconfig_address.
reconfig_readdata_ch<N-1>[7:0], ...,reconfig_readdata_ch0[7:0]	Output	reconfig_clk_ch#	A 8-bit data read bus for each channel. Valid data is placed on this bus after a read operation. Signal is valid after reconfig_waitrequest goes high and then low.
reconfig_waitrequest_ch<N-1>, ..., reconfig_waitrequest_ch0	Output	reconfig_clk_ch#	A one-bit signal for each channel that indicates that the AVMM interface is busy. Keep the AVMM command asserted until the interface is ready to proceed with the read/write transfer.

Table 67. AVMM Interface Parameters

The following parameters are available in the **Dynamic Reconfiguration** tab of the Transceiver Native PHY parameter editors.

Parameter	Value	Description
Share reconfiguration interface	On/Off	Enables you to use a single reconfiguration interface to control all channels. Off by default. If enabled, the uppermost bits of reconfig_address identifies the active channel. The lower 20 bits specify the reconfiguration address. Binary encoding is used to identify the active channel (available only for Transceiver Native PHY). Enable this option, if desired, when the Native PHY is configured with more than one channel.
Enable dynamic reconfiguration	On/Off	Enables the reconfiguration interface. Off by default. The reconfiguration interface is exposed when this option is enabled.
Enable Native PHY Debug Master Endpoint	On/Off	When enabled, the Native PHY Debug Master Endpoint (NPDME) is instantiated and has access to the AVMM interface of the Native PHY. You can access certain test and debug functions using System Console with the NPDME. Refer to the "Embedded Debug Features" section for more details about NPDME.

continued...

Parameter	Value	Description
Enable capability registers	On/Off	Enables capability registers. These registers provide high-level information about the transceiver channel's configuration.
Set user-defined IP identifier	User-defined	Sets a user-defined numeric identifier that can be read from the `user_identifier` offset when the capability registers are enabled.
Enable control and status registers	On/Off	Enables soft registers for reading status signals and writing control signals on the PHY interface through the NPDME or reconfiguration interface.
Configuration file prefix	User-defined	Specifies the file prefix used for generating configuration files. Use a unique prefix for configuration files for each variant of the Native PHY.
Generate SystemVerilog package File	On/Off	Creates a SystemVerilog package file that contains the current configuration data values for all reconfiguration addresses. Disabled by default.
Generate C header file	On/Off	Creates a C header file that contains the current configuration data values for all reconfiguration addresses. Disabled by default.
Generate MIF (Memory Initialize File)	On/Off	Creates a MIF file that contains the current configuration data values for all reconfiguration addresses. Disabled by default.
Enable multiple reconfiguration profiles	On/Off	Use the **Parameter Editor** to store multiple configurations. The parameter settings for each profile are tabulated in the **Parameter Editor**.
Enable embedded reconfiguration streamer	On/Off	Embeds the reconfiguration streamer into the Native PHY IP core, and automates the dynamic reconfiguration process between multiple predefined configuration profiles.
Generate reduced reconfiguration files	On/Off	Enables the Native PHY IP core to generate reconfiguration files that contain only the attributes that differ between multiple profiles.
Number of reconfiguration profiles	1 to 8	Specifies the number of reconfiguration profiles to support when multiple reconfiguration profiles are enabled.
Store current configuration to profile	0 to 7	Selects which reconfiguration profile to store.
Store configuration to selected profile	N/A	Stores the current Native PHY parameter settings to the profile specified by **Store current configuration to profile** specified.
Load configuration from selected profile	N/A	Loads the current Native PHY parameter settings to the profile specified by **Store current configuration to profile** specified.

continued...

Parameter	Value	Description
Clear selected profile	N/A	Clears the stored Native PHY parameter settings for the profile specified by the **Store current reconfiguration to profile** parameter. An empty profile defaults to the current parameter settings of the Native PHY. In other words, an empty profile reflects the Native PHY current parameter settings.
Clear all profiles	N/A	Clears the Native PHY IP parameter settings for all profiles.
Refresh selected profile	N/A	Equivalent to clicking the **Load configuration from selected profile** and **Store configuration to selected profile** buttons in sequence. This operation loads the parameter settings from stored profile specified by the **Store current configuration to profile** parameter and then stores the parameters back to the profile.

7.14. Embedded Debug Features

The Native PHY IP cores provide the following optional debug features to facilitate embedded test and debug capability:

- Native PHY Debug Master Endpoint (NPDME)
- Optional Reconfiguration Logic

7.14.1. Native PHY Debug Master Endpoint (NPDME)

The NPDME is a JTAG-based AVMM master that provides access to the transceiver registers through the system console. You can enable NPDME using the **Enable Native PHY Debug Master Endpoint** option available under the **Dynamic Reconfiguration** tab in the Native PHY IP cores. When using NPDME, the Intel Quartus Prime software inserts the debug interconnect fabric to connect with USB, JTAG, or other net hosts. Select the **Share Reconfiguration Interface** parameter when the Native PHY IP instance has more than one channel. The Transceiver Toolkit, a useful tool in debugging transceiver links, requires NPDME.

When you enable NPDME in your design, you must do one of the following:

- Connect an AVMM master to the reconfiguration interface.
- Connect the `reconfig_clk`, `reconfig_reset` signals and ground the `reconfig_write`, `reconfig_read`, `reconfig_address`, and `reconfig_write` data signals of the reconfiguration interface if not being driven by other core logic. If you do not connect the reconfiguration interface signals appropriately, the NPDME has no clock or reset and functions unexpectedly. Refer to the example connection below (this is an example of a single channel with no internal logic driving the reconfig interface):

```
.reconfig_clk (mgmt_clk),
.reconfig_reset (mgmt_reset),
.reconfig_write (1'b0),
.reconfig_address (19'b0),
.reconfig_read (1'b0),
.reconfig_writedata (8'b0),
```

7.14.2. Optional Dynamic Reconfiguration Logic

The E-tile Transceiver Native PHY IP cores contain soft logic for debug purposes known as the Optional Reconfiguration Logic. This soft logic provides a set of registers that enable you to determine the state of the Native PHY IP cores.

You can enable the following optional reconfiguration logic options in the transceiver Native PHY IP cores:

- Capability registers
- Control and status registers

7.14.2.1. Capability Registers

The capability registers provide high level information about the transceiver channel configuration and capture a set of chosen capabilities of the PHY that cannot be reconfigured. They are located on the PMA AVMM interface and are located on addresses 0x40000 to 0x5FFFF.

Related Information

PMA Register Map on page 211

7.14.2.2. Control and Status Registers

Control and status registers are optional registers that memory map the status outputs from and control inputs to the Native PHY. The control and status registers are located on the PMA AVMM interface from 0x40000 to 0x5FFFF.

Related Information

PMA Register Map on page 211

7.15. Timing Closure Recommendations

Intel recommends that you enable the multiple reconfiguration profiles feature in the E-tile Native PHY IP core if any of the modified or target configurations involve changes to RS-FEC settings. Using multiple reconfiguration profiles is optional if the reconfiguration involves changes to only PMA settings such as TX Attenuation Value (VOD) swing or refclk switching. When performing a dynamic reconfiguration, you must:

- Include constraints to create the extra clocks for all modified or target configurations at the RS-FEC -FPGA fabric interface. Clocks for the base configuration are created by the Intel Quartus Prime software. These clocks enable the Intel Quartus Prime Pro Edition to perform static timing analysis for all the transceiver configurations and their corresponding FPGA fabric core logic blocks.
- Include the necessary false paths between the RS-FEC – FPGA fabric interface and the core logic.

For example, you can perform dynamic reconfiguration to switch the datapath from PMA direct to RS-FEC using the multiple reconfiguration profiles feature.

7.16. Transceiver Register Map

The transceiver register map provides a list of available PCS, PMA, and EMIB addresses that are used in the reconfiguration process (the transceiver configuration file includes details about the registers that are set for a specific transceiver configuration). Do not use the register map to locate and modify specific registers in the transceiver. Doing so may result in an illegal configuration. Refer to a valid transceiver configuration file for legal register values and combinations.

7.17. Loading IP Configuration Settings

You may need to set the PMA RX adaptation settings for optimal transceiver performance. See the *PMA Tuning* for more information. There are two methods for setting the PMA RX adaption values.

- The first method is to use PMA attributes to load the PMA RX adaptation settings one by one. See *PMA Attribute Codes* and *Configuring a PMA Parameter Tunable by the Adaptive Engine* for details of how to load an adaptation setting.
- The second method is, when you generate the IP, to define up to eight different PMA configurations in the Native PHY IP to use built-in logic to load one of them to all transceiver channels in the instance at run-time. See*Loading IP Configuration Settings Process* for more details and *Configuring a PMA Parameter Using Native PHY IP* for an example.

Related Information

- Loading IP Configuration Settings on page 176
- PMA Tuning on page 81
- Loading IP Configuration Settings Process on page 176
- Configuring a PMA Parameter Tunable by the Adaptive Engine on page 195
- PMA Attribute Codes on page 216

7.17.1. Loading IP Configuration Settings Process

1. Use the control and status register 0x40143 to load a profile into the PMA. Register 0x40143[2:0] selects which configuration to load and 0x40143[7] indicates to the built-in logic to do the actual load.
2. Poll register 0x40144[0] until it becomes 1 to indicate the configuration loading has completed.
3. Set register 0x200 to register 0x203 as indicated in *Loading Parameters into the Receiver*.
4. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.

Related Information

Loading Parameters into the Receiver on page 232

7.17.2. Alternative Method for Setting PMA Attributes

In addition to loading PMA attributes one by one, you can use registers 0x200 to 0x203 to load a set of common PMA attributes to all transceiver channels in the instance. Please refer to *Reading and Writing PMA Analog Parameters Using Attributes* for more details. Refer to *Configuring a PMA Parameter Using Native PHY IP* for an example.

Related Information

- Configuring a PMA Parameter Using Native PHY IP on page 198
- Reading and Writing PMA Analog Parameters Using Attributes on page 231

7.18. Dynamic Reconfiguration Revision History

Document Version	Changes
2019.10.11	Made the following changes: • Changed the reference clock frequency range is from 125 MHz to 500 MHz to 125 MHz to 700 MHz. • Added more instructions for *Changing the Reference Clock Frequency on* `refclk[0, 2, 3, 4, 5, 6, 7, 8]`. • Added the Related Information links for the Intel Agilex device documents.
2019.07.29	Made the following changes: • Updated the *Switching Reference Clocks* steps. • Added procedures for the correct dynamic reconfiguration on reference clocks.
2019.04.19	Made the following changes: • Added *Register 0x40140*. • Added *Register 0x40141*.
2019.02.04	Made the following changes: • Added *Loading IP Configuration Settings*. • Added *Loading IP Configuration Settings Process*. • Added *Alternative Method for Setting PMA Attributes*.
2018.10.08	Made the following changes: • Changed some descriptions in the "Dynamic Reconfiguration" section. • Added a feature to the "Intel Stratix 10 Dynamic Reconfiguration Feature Support" table. • Updated the reconfig_write waveform in the "Writing to the Reconfiguration Interface" figure. • Switched the order of Disable PMA internal serial loopback and Enable Initial RX Equalizer steps in the "Dynamic Reconfiguration with Native PHY in Automatic Reset Mode" figure. • Changed Step 4 in the "Embedded Reconfiguration Streamer" section. • Added step 8f and 9 to *Switching Reference Clocks*. • Added a reference to *PMA Analog Reset* from *Switching Reference Clocks*.
2018.07.18	Made the following changes: • Added reconfiguring the PMA between NRZ and PAM4 (non high data rate) modes as a feature to the "Dynamic Reconfiguration Feature Support" table. • Added *Multiple Reconfiguration Profiles*. • Updated *Reconfiguration Files* with entirely new information. • Added *Embedded Reconfiguration Streamer*. • Updated the steps in the *Switching Reference Clocks* section.
2018.05.15	Made the following changes:

continued...

Document Version	Changes
	• Removed "Configuration Files," "Multiple Dynamic Reconfiguration Profiles," "Changing Analog PMA Settings," "Multiple Dynamic Reconfiguration Profiles," "Embedded Dynamic Reconfiguration Streamer," "PMA Attribute Sequencer," and "Native PHY IP Core or Clocking Resources Guided Reconfiguration Flow" sections. • Added RX adaptation to the list of PMA analog features. • Removed reconfig_rsfec pending support. • Removed "PCS" from all sections. • Updated MIF file format in the "Configuration Files" section. • Updated the configuration file location in the "Multiple Reconfiguration Profiles" section. • Removed the "Register read/write sequencer" block from the "Arbitration" figure and list of programmable registers. • Deleted data-rate-non-specific recommendation from the "Recommendations for Dynamic Reconfiguration" section. • Replaced the text in the "Steps to Perform Dynamic Reconfiguration" section with "Dynamic Reconfiguration with Reset Controller in Automatic Mode" and "Dynamic Reconfiguration with Reset Controller in Manual Mode (required for fractured RS-FEC mode)." • Changed "Direct Reconfiguration Flow" with "PMA Register Read/Write Details." • Removed references to the "Steps to Perform Dynamic Reconfiguration" section. • Added steps to reset and reconfigure the PMA in the "Switching Reference Clocks" section. • Removed reference to the PMA register read/write sequencer for this release pending testing in the "Changing Analog PMA Settings" section. • Updated the address, writedata, and readdata bus widths in the "Ports and Parameters" and "NPDME" sections.
2018.01.31	Initial release.

8. Dynamic Reconfiguration Examples

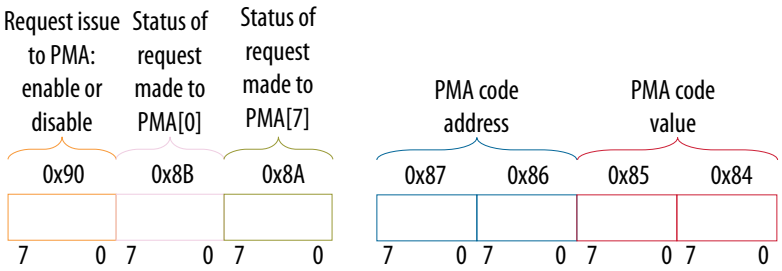
8.1. Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode

Dynamic reconfiguration is the process of modifying transceiver channels to meet changing requirements during device operation. You can customize channels by initiating reconfiguration during device operation or following power-up.

Common PMA parameters that are reconfigured are the reference clock source, data rate and PMA serialization/deserialization factor. Refer to *PMA Attribute Codes* for how to change the attribute codes. The following example shows how to reconfigure a duplex PMA channel in NRZ mode in order to:

- Reconfigure the PMA reference clock from `refclk0` to `refclk1` as listed in the Native PHY IP
- Change the PMA baud rate
- Change the deserialization factor

Figure 105. Sending PMA Attribute Through AVMM Registers

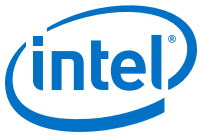


Reconfiguration Flow

1. Assert `tx_reset/rx_reset`.
2. Wait for the `tx_ready/rx_ready` to deassert.
3. Disable the PMA by using PMA attribute code 0x0001.
   a. Optional: Write 0x8A[7] = 0x1 to ensure the PMA attribute status flag (0x8A[7] for the previous attribute) is cleared before writing to registers 0x84 to 0x87 to load in the new PMA attribute.
   b. Write 0x84[7:0] = 0x00.
   c. Write 0x85[7:0] = 0x00.
   d. Write 0x86[7:0] = 0x01.
   e. Write 0x87[7:0] = 0x00.
   f. Write 0x90[0] = 1'b1.

g. Read 0x8A[7]. It should be 1.
h. Read 0x8B[0] until it changes to 0.
i. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.

4. Wait for `tx_pma_ready/rx_pma_ready` to deassert.
5. Refer to *Switching Reference Clocks* for how to change the reference clock.
6. Reset the internal controller inside the PMA because the REFCLK source changed by:
   a. Write 0x200[7:0] = 0x00.
   b. Write 0x201[7:0] = 0x00.
   c. Write 0x202[7:0] = 0x00.
   d. Write 0x203[7:0] = 0x81.
   e. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.
7. Change TX/RX baud rate to refclk * 50 and above 15 Gbps by using PMA attribute code 0x0005.
   a. Write 0x84[7:0] = 0x32.
   b. Write 0x85[7:0] = 0x80 (bit 7 applies the update to both TX/RX).
   c. Write 0x86[7:0] = 0x05.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
8. Change serialization/deserialization factor to 40 bits wide by using PMA attribute code 0x0014.
   a. Write 0x84[7:0] = 0x33.
   b. Write 0x85[7:0] =0x00.
   c. Write 0x86[7:0] = 0x14.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
9. Change the RX phase slip to maximize the RX PMA timing margin.
   a. Write 0x84[7:0] = 0x00.
   b. Write 0x85[7:0] = 0xA4 to slip by 36 UI.
   c. Write 0x86[7:0] = 0x0E.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.



f. Read 0x8A[7]. It should be 1.
g. Read 0x8B[0] until it changes to 0.
h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.

10. Enable the PMA by using PMA attribute code 0x0001.
    a. Write 0x84[7:0] = 0x07.
    b. Write 0x85[7:0] =0x00.
    c. Write 0x86[7:0] = 0x01.
    d. Write 0x87[7:0] = 0x00.
    e. Write 0x90[0] = 1’b1.
    f. Read 0x8A[7]. It should be 1.
    g. Read 0x8B[0] until it changes to 0.
    h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
11. Enable PRBS31 data.
    a. Write 0x84[7:0] = 0x25.
    b. Write 0x85[7:0] = 0x03.
    c. Write 0x86[7:0] = 0x02.
    d. Write 0x87[7:0] = 0x00.
    e. Write 0x90[0] = 1'b1.
    f. Read 0x8A[7]. It should be 1.
    g. Read 0x8B[0] until it changes to 0.
    h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
12. Change to internal serial loopback mode by using PMA attribute code 0x0008.
    a. Write 0x84[7:0] = 0x01.
    b. Write 0x85[7:0] = 0x01.
    c. Write 0x86[7:0] = 0x08.
    d. Write 0x87[7:0] = 0x00.
    e. Write 0x90[0] = 1’b1.
    f. Read 0x8A[7]. It should be 1.
    g. Read 0x8B[0] until it changes to 0.
    h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
13. Enable initial coarse adaptive equalization by using PMA attribute code 0x000A.
    a. Write 0x84[7:0] = 0x01.
    b. Write 0x85[7:0] = 0x00.
    c. Write 0x86[7:0] = 0x0A.
    d. Write 0x87[7:0] = 0x00.
    e. Write 0x90[0] = 1'b1.
    f. Read 0x8A[7]. It should be 1.
    g. Read 0x8B[0] until it changes to 0.

h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.

14. Read initial coarse adaptation status by using PMA attribute code 0x0126.
    a. Write 0x84[7:0] = 0x00.
    b. Write 0x85[7:0] = 0x0B.
    c. Write 0x86[7:0] = 0x26.
    d. Write 0x87[7:0] = 0x01.
    e. Write 0x90[0] = 1'b1.
    f. Read 0x8A[7]. It should be 1.
    g. Read 0x8B[0] until it changes to 0.
    h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
    i. Read register 0x88. Repeat step 12 until 0x88[0] has pulsed high, followed by a pulse low to indicate the initial coarse adaptation has completed.
15. Load PMA analog configuration if applicable.
    a. Write 0x40143 = 0x80 plus the configuration number you want to load. For example, write 0x82 to load configuration 2.
    b. Read 0x40144[0] until it reports 0x1.

Skip the next step if you are running internal serial loopback mode.

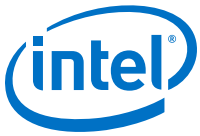
16. Enable mission mode, or disable internal serial loopback mode.
    a. Write 0x200 = 0xD2.
    b. Write 0x201 = 0x00.
    c. Write 0x202 = 0x00.
    d. Write 0x203 = 0x96. This picks the opcode for START_ADAPTATION.
    e. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.

Skip the next step if you are running internal serial loopback mode.

17. Enable initial coarse adaptive equalization in mission mode by using PMA attribute code 0x000A.
    a. Write 0x200 = 0xD2.
    b. Write 0x201 = 0x00.
    c. Write 0x202 = 0x01.
    d. Write 0x203 = 0x96. This picks the opcode for START_ADAPTATION.
    e. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.

Skip the next step if you are running internal serial loopback mode.

18. Read initial coarse adaptation status by using PMA attribute code 0x0126.
    a. Write 0x84[7:0] = 0x00.
    b. Write 0x85[7:0] = 0x0B.
    c. Write 0x86[7:0] = 0x26.
    d. Write 0x87[7:0] = 0x01.
    e. Write 0x90[0] = 1'b1.



f. Read 0x8A[7]. It should be 1.
g. Read 0x8B[0] until it changes to 0.
h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
i. Read register 0x88. Repeat this step until 0x88[0] has pulsed high, followed by a pulse low to indicate the initial coarse adaptation has completed.

19. Deassert `tx_reset/rx_reset`.
20. Disable PRBS31 data.
    a. Write 0x200 = 0xE2.
    b. If in mission mode, write 0x201 = 0x01.
    c. If in internal serial loopback mode, write 0x201 = 0x03.
    d. Write 0x202 = 0x00.
    e. Write 0x203 = 0x96. This picks the opcode for START_ADAPTATION.
    f. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.
21. If valid data rate traffic is available at RX, go to the next step, or re-run initial coarse adaptation until the traffic is valid.
22. If you need to set a PMA analog parameter before running continuous adaptation, send the attributes. Refer to *Reading and Writing PMA Analog Parameters Using Attributes*.
23. Enable continuous adaptive equalization by using PMA attribute code 0x000A.
    a. Write 0x84[7:0] = 0x03.
    b. Write 0x85[7:0] = 0x00.
    c. Write 0x86[7:0] = 0x0A.
    d. Write 0x87[7:0] = 0x00.
    e. Write 0x90[0] = 1'b1.
    f. Read 0x8A[7]. It should be 1.
    g. Read 0x8B[0] until it changes to 0.
    h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
24. Optionally, check the `rx_is_lockedtodata` output pin.
25. The link is up.

Related Information

- Switching Reference Clocks on page 161
- PMA Register Map on page 211
- PMA Attribute Codes on page 216
- Reading and Writing PMA Analog Parameters Using Attributes on page 231

8.2. PRBS Usage Model

The PRBS usage model is comprised of the PRBS pattern generator and verifier configuration and the hard PRBS error counter configuration. You can use PMA Direct Mode to implement PRBS (10G/25G) channels.

Different PRBS patterns can be configured using the 0x84, 0x85, 0x86, and 0x87 AVMM addresses. The 0x84 and 0x85 AVMM addresses point to the PRBS pattern code. The 0x86 and 0x87 AVMM addresses point to the PMA code address 0x02.

Figure 106. Setting PMA Attributes for PRBS Through AVMM Registers

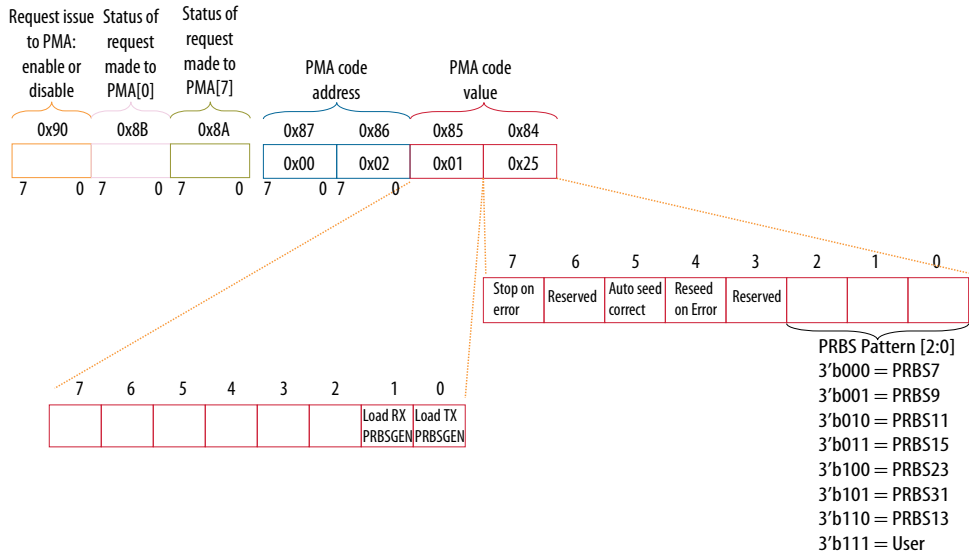


Table 68. PRBS Control PMA Attribute Code Definition, PMA Attribute Code 0x02, PRBS Enable

Address	Direction	Definition
0x84[2:0]	input	3'b000: prbs7 3'b001: prbs9 3'b010: prbs11 3'b011: prbs15 3'b100: prbs23 3'b101: prbs31 3'b110: prbs13 3'b111: user
0x84[4]	input	Reseed on error
0x84[5]	input	Autoseed correct (generator goes from all '0' to all '1' if it occurs)
0x84[7]	input	Stop on error (RX)
0x85[0]	input	Load TX PRBSGEN
0x85[1]	input	Load RX PRBSGEN

Table 69. PRBS Control PMA Attribute Code Definition, PMA Attribute Code 0x02, PRBS Disable

Address	Direction	Definition
0x85[7:0] 0x84[7:0]	input	Disable codes 0x3ff: disable both generators 0x1ff: disable TX PRBSGEN

continued...

Address	Direction	Definition
		0x2ff: disable RX PRBSGEN
0x89[7:0] 0x88[7:0]	return value	0x00: Failed due to background processes needing time to complete operations that may change the requested configuration. Wait some time and re-issue the request. 0x02: Success

For example, to use the PRBS31 generator and checker, do the following steps:

1. Set TX PRBS31.
   a. Write 0x84[7:0] = 0x25.
   b. Write 0x85[7:0] = 0x01.
   c. Write 0x86[7:0] = 0x02.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
2. Set RX PRBS31.
   a. Write 0x84[7:0] = 0x35.
   b. Write 0x85[7:0] = 0x02.
   c. Write 0x86[7:0] = 0x02.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
3. Enable the transceiver channel if it is not running already.
   a. Write 0x84[7:0] = 0x07.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x01.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
4. Wait for `tx_ready` and `rx_ready` to both be 1.
5. Set the data comparator.

a. Write 0x84[7:0] = 0x03.
b. Write 0x85[7:0] = 0x02.
c. Write 0x86[7:0] = 0x03.
d. Write 0x87[7:0] = 0x00.
e. Write 0x90[0] = 1'b1.
f. Read 0x8A[7]. It should be 1.
g. Read 0x8B[0] until it changes to 0.
h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.

6. Reset error counters.
   a. Write 0x84[7:0] = 0x00.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x17.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
7. Wait for the 32 bits wide error counter to be accumulated.
8. Set the error count to be read out.
   a. Write 0x84[7:0] = 0x03.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x18.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
9. Read the lower 16 bits of the error counter.
   a. Write 0x84[7:0] = 0x00.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x1A.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.

h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
i. Read 0x88[7:0]. This represents bits [7:0] of the error counter.
j. Read 0x89[7:0]. This represents bits [15:8] of the error counter.

10. Read the upper 16 bits of the error counter.
    a. Write 0x84[7:0] = 0x00.
    b. Write 0x85[7:0] = 0x00.
    c. Write 0x86[7:0] = 0x1a.
    d. Write 0x87[7:0] = 0x00.
    e. Write 0x90[0] = 1'b1.
    f. Read 0x8A[7]. It should be 1.
    g. Read 0x8B[0] until it changes to 0.
    h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
    i. Read 0x88[7:0]. This represents bits [23:16] of the error counter.
    j. Read 0x89[7:0]. This represents bits [31:24] of the error counter.

Note: During PMA performance verification testing, with continuous adaptation running in background, error bits cannot be accumulated to calculate BER because the Hard PRBS error counter is in a busy state. You can read errors during continuous adaptation by implementing a soft PRBS generator and verifier. Errors can be accumulated in hard PRBS error counter after stopping the continuous adaptation.

8.3. PMA Error Injection

Inject either a single error or a burst of errors on the TX driver output using PMA attribute codes. This switches the internal TX error injection signal on and off for the number of bits requested.

Figure 107. Injecting Error Bits Using the PMA Attributes Through AVMM Registers

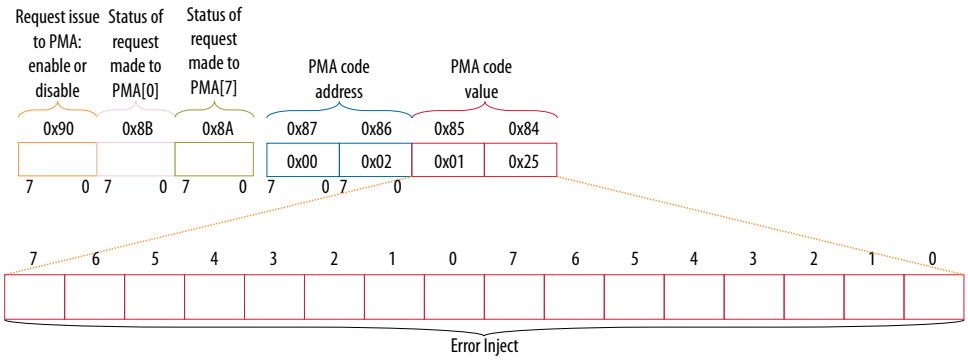


Table 70. PMA Attribute Code Error Inject Bits

Name	Address	Bit Range	Description
Error Inject	0x001B	0x85[7:0], 0x84[7:0]	Number of errors to inject

1. Set Error Injection for injecting single bit error.

a. Write 0x84[7:0] = 0x01.
b. Write 0x85[7:0] = 0x00.
c. Write 0x86[7:0] = 0x1B.
d. Write 0x87[7:0] = 0x00.
e. Write 0x90[0] = 1’b1.
f. Read 0x8A[7]. It should be 1.
g. Read 0x8B[0] until it changes to 0.
h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.

2. Set Error Injection for injecting burst of 10 bit errors.
   a. Write 0x84[7:0] = 0x0A.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x1B.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1’b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.

8.4. PMA Receiver Equalization Adaptation Usage Model

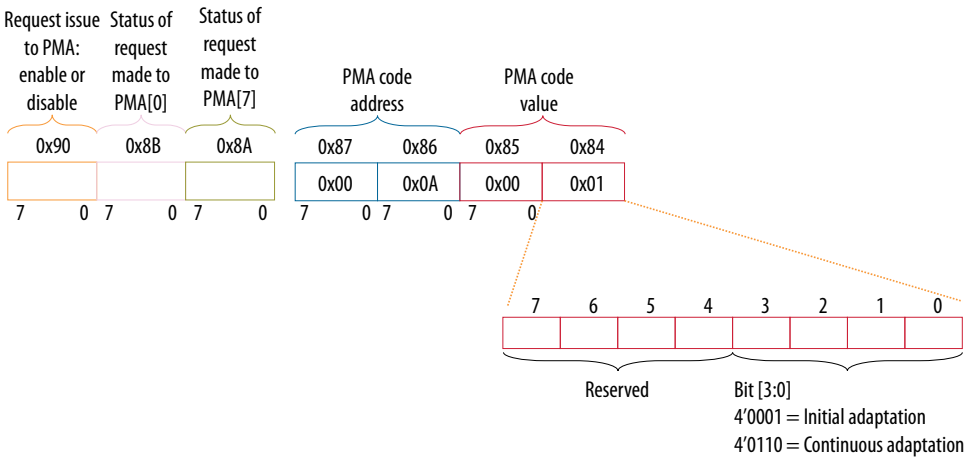
The PMA receiver adaptive equalization engine allows the equalization blocks to adapt to an optimal value. These optimal values can be read back.

Adaptive equalization includes the following modes:

- Initial adaptation
- Continuous adaptation

The PMA attribute programming steps are as follows. Refer to the register map to configure the PMA receiver adaptive equalization modes.

Figure 108. Equalizer Bits



Initial Adaptation

The procedure is as follows:

1. Configure PMA attribute code 0x01 as following to enable transmitter and receiver:
   a. Write 0x84[7:0] = 0x07.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x01.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
2. Configure PMA attribute code 0x0A as following to enable initial adaptation:
   a. Write 0x84[7:0] = 0x01.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x0A.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
3. Read the initial adaptation equalization status:
   a. Write 0x84[7:0] = 0x00.
   b. Write 0x85[7:0] = 0x0B.
   c. Write 0x86[7:0] = 0x26.
   d. Write 0x87[7:0] = 0x01.

e. Write 0x90[0] = 1'b1.
f. Read 0x8A[7]. It should be 1.
g. Read 0x8B[0] until it changes to 0.
h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
i. Read 0x88[0].

4. Repeat Step 4 until 0x88[0] goes from 1 to 0.

Continuous Adaptation

1. Configure PMA attribute code 0x0A as following to enable continuous adaptation:
   a. Write 0x84[7:0] = 0x06.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x0A.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.

Note: You can stop continuous adaptation. Refer to *Receiver Tuning Controls* for more information.

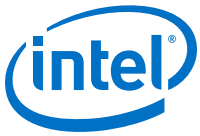
Related Information

0x000A: Receiver Tuning Controls on page 221

8.5. User-Defined Pattern Example

This works for patterns such as a clock pattern (001100110011…) on the PRBS generator in the PMA.

1. Disable the PMA TX output.
   a. Write 0x84[7:0] = 0x03.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x01.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1’b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
2. Select the TX data with a status/debug register.
   a. Write 0x84[7:0] = 0x00.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x18.

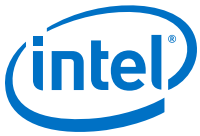


d. Write 0x87[7:0] = 0x00.
e. Write 0x90[0] = 1'b1.
f. Read 0x8A[7]. It should be 1.
g. Read 0x8B[0] until it changes to 0.
h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

3. Load pattern [9:0].
   a. Write 0x84[7:0] = 0x33.
   b. Write 0x85[7:0] = 0x03.
   c. Write 0x86[7:0] = 0x19.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
4. Load pattern [19:10].
   a. Write 0x84[7:0] = 0xCC.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x19.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
5. Load pattern [29:20].
   a. Write 0x84[7:0] = 0x33.
   b. Write 0x85[7:0] = 0x03.
   c. Write 0x86[7:0] = 0x19.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
6. Load pattern [39:30].
   a. Write 0x84[7:0] = 0xCC.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x19.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.

f. Read 0x8A[7]. It should be 1.
g. Read 0x8B[0] until it changes to 0.
h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

7. Load pattern [49:40].
   a. Write 0x84[7:0] = 0x33.
   b. Write 0x85[7:0] = 0x03.
   c. Write 0x86[7:0] = 0x19.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
8. Load pattern [59:50].
   a. Write 0x84[7:0] = 0xCC.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x19.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
9. Load pattern [69:60].
   a. Write 0x84[7:0] = 0x33.
   b. Write 0x85[7:0] = 0x03.
   c. Write 0x86[7:0] = 0x19.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
10. Load pattern [79:70].
    a. Write 0x84[7:0] = 0xCC.
    b. Write 0x85[7:0] = 0x00.
    c. Write 0x86[7:0] = 0x19.
    d. Write 0x87[7:0] = 0x00.
    e. Write 0x90[0] = 1'b1.
    f. Read 0x8A[7]. It should be 1.
    g. Read 0x8B[0] until it changes to 0.



h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

11. Load the TX PRBS generator with your pattern.
    a. Write 0x84[7:0] = 0x27.
    b. Write 0x85[7:0] = 0x01.
    c. Write 0x86[7:0] = 0x02.
    d. Write 0x87[7:0] = 0x00.
    e. Write 0x90[0] = 1'b1.
    f. Read 0x8A[7]. It should be 1.
    g. Read 0x8B[0] until it changes to 0.
    h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
12. Enable the PMA TX output.
    a. Write 0x84[7:0] = 0x07.
    b. Write 0x85[7:0] = 0x00.
    c. Write 0x86[7:0] = 0x01.
    d. Write 0x87[7:0] = 0x00.
    e. Write 0x90[0] = 1'b1.
    f. Read 0x8A[7]. It should be 1.
    g. Read 0x8B[0] until it changes to 0.
    h. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

8.6. Configuring the Attenuation Value (VOD)

This usage model covers configuring attenuation value (VOD) of 0x1 for the PMA driver.

Setting the attenuation parameter (VOD) which is configurable in the TX Equalizer block

1. Write 0x84[7:0] = 0x01.
2. Write 0x85[7:0] = 0x40.
3. Write 0x86[7:0] = 0x15.
4. Write 0x87[7:0] = 0x00.
5. Write 0x90[0] = 1'b1.
6. Read 0x8A[7]. It should be 1.
7. Read 0x8B [0] until it changes to 0.
8. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
9. Read 0x8A[7]. It should be 0.

8.7. Configuring the Post Emphasis Value

This usage model covers configuring the Post Emphasis value in the TX Equalizer for decimal values 2 and -2. The negative decimal value is represented as an 8-bit 2's complement.

Example 1: Setting the post_tap emphasis of the TX Equalizer with PMA attribute values of 2 (decimal) and 0x02 (hexadecimal) or -2 (decimal) and 0xFE (hexadecimal) and PMA attribute code of 0x15

1. Write 0x84[7:0] = 0x02 for 2 (decimal), or write 0x84[7:0] = 0xFE for -2 (decimal).
2. Write 0x85[7:0] = 0x80.
3. Write 0x86[7:0] = 0x15.
4. Write 0x87[7:0] = 0x00.
5. Write 0x90[0] = 1'b1.
6. Read 0x8A[7]. It should be 1.
7. Read 0x8B [0] until it changes to 0.
8. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
9. Read 0x8A[7]. It should be 1.

Example 2: Setting the post_tap emphasis of the TX Equalizer with PMA attribute values of -18 (decimal) and 0xEE (2's complement) and PMA attribute code of 0x15

10. Write 0x84[7:0] = 0xEE.
11. Write 0x85[7:0] = 0x80.
12. Write 0x86[7:0] = 0x15.
13. Write 0x87[7:0] = 0x00.
14. Write 0x90[0] = 1'b1.
15. Read 0x8A[7]. It should be 1
16. Read 0x8B [0] until it changes to 0.
17. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
18. Read 0x8A[7]. It should be 1.

8.8. Configuring `pretap1` Values

This usage model covers configuring `pretap1` values of 10 (decimal) and 0x0A (hexadecimal) in the TX Equalizer.

1. Write 0x84[7:0] = 0x0A.
2. Write 0x85[7:0] = 0x00.
3. Write 0x86[7:0] = 0x15.
4. Write 0x87[7:0] = 0x00.
5. Write 0x90[0] = 1'b1.
6. Read 0x8A [7]. It should be 1.
7. Read 0x8B [0] until it changes to 0.
8. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

8.9. Inverting TX Polarity for the PMA Driver

This procedure inverts the TX polarity.

1. Write 0x84[7:0] = 0x01.
2. Write 0x85[7:0] = 0x03.
3. Write 0x86[7:0] = 0x13.
4. Write 0x87[7:0] = 0x00.
5. Write 0x90[0] = 1'b1.
6. Read 0x8A[7]. It should be 1.
7. Read 0x8B[0] until it changes to 0.
8. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

8.10. Inverting RX Polarity for the PMA Driver

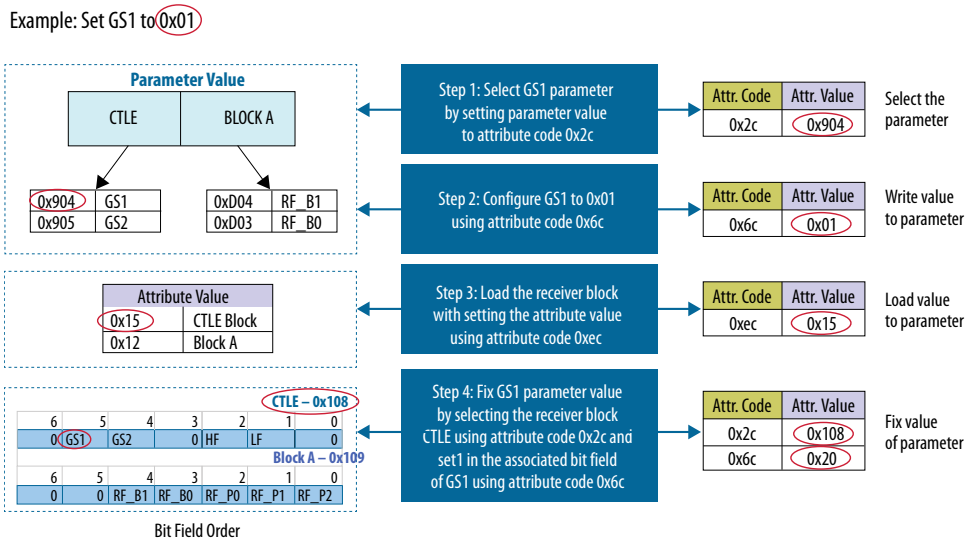
This procedure inverts the RX polarity.

1. Write 0x84[7:0] = 0x10.
2. Write 0x85[7:0] = 0x03.
3. Write 0x86[7:0] = 0x13.
4. Write 0x87[7:0] = 0x00.
5. Write 0x90[0] = 1'b1.
6. Read 0x8A[7]. It should be 1.
7. Read 0x8B[0] until it changes to 0.
8. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

8.11. Configuring a PMA Parameter Tunable by the Adaptive Engine

This section provides details on how you can set a PMA parameter to a fixed value, making it no longer adaptable.Most of the PMA parameters by default are tuned by the RX adaptation engine modes. However, they can be manually configured by assigning values to them; if you assign a value to a PMA parameter, it can no longer be updated by the adaptation engine.

Figure 109. Example Use Case to Configure PMA Parameters



1. Configure the GS1 PMA parameter to 0x01.

Select the PMA parameter by setting the PMA attribute code 0x2C to PMA attribute value 0x904.

2. Write 0x84[7:0] = 0x04.
3. Write 0x85[7:0] = 0x09.
4. Write 0x86[7:0] = 0x2C.
5. Write 0x87[7:0] = 0x00.
6. Write 0x90[0] = 1'b1.
7. Read 0x8A[7]. It should be 1.
8. Read 0x8B[0], until it changes to 0.
9. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

Write a value to the PMA parameter by setting the PMA attribute code 0x6C to PMA attribute value 0x01.

10. Write 0x84[7:0] = 0x01.
11. Write 0x85[7:0] = 0x00.
12. Write 0x86[7:0] = 0x6C.
13. Write 0x87[7:0] = 0x00.
14. Write 0x90[0] = 1'b1.
15. Read 0x8A[7]. It should be 1.
16. Read 0x8B[0], until it changes to 0.
17. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

Load the value to the PMA parameter by setting the PMA attribute code 0xEC to PMA attribute value 0x15.

18. Write 0x84[7:0] = 0x15.
19. Write 0x85[7:0] = 0x00.

20. Write 0x86[7:0] = 0xEC.
21. Write 0x87[7:0] = 0x00.
22. Write 0x90[0] = 1'b1.
23. Read 0x8A[7]. It should be 1.
24. Read 0x8B[0], until it changes to 0.
25. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

Set the PMA parameter so that it is not overwritten by the adaptive tuning engine by configuring the PMA attribute code 0x2C to attribute value 0x108 and PMA attribute code 0x6C to attribute value 0x20.

26. Write 0x84[7:0] = 0x08.
27. Write 0x85[7:0] = 0x01.
28. Write 0x86[7:0] = 0x2C.
29. Write 0x87[7:0] = 0x00.
30. Write 0x90[0] = 1'b1.
31. Read 0x8A[7]. It should be 1.
32. Read 0x8B[0], until it changes to 0.
33. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.
34. Write 0x84[7:0] = 0x20.
35. Write 0x85[7:0] = 0x00.
36. Write 0x86[7:0] = 0x6c.
37. Write 0x87[7:0] = 0x00.
38. Write 0x90[0] = 1'b1.
39. Read 0x8A[7]. It should be 1.
40. Read 0x8B[0], until it changes to 0.
41. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

    Refer to the *Register Map* for more details.

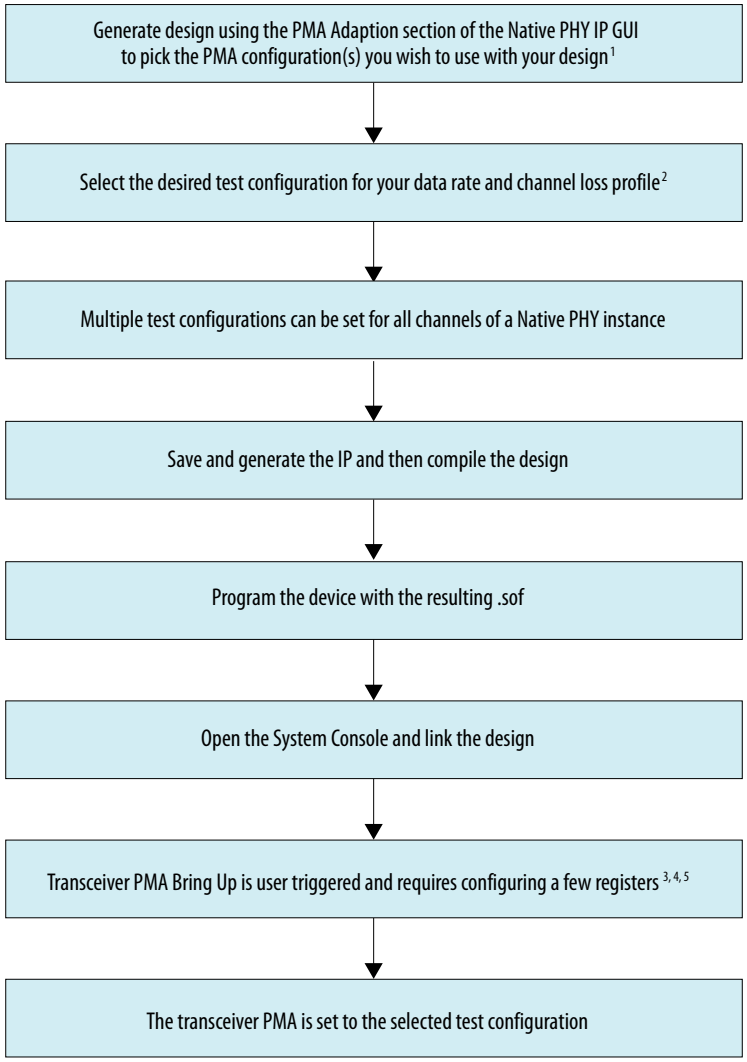
Related Information

Register Map on page 211

8.12. Configuring a PMA Parameter Using Native PHY IP

8.12.1. PMA Bring Up Flow Using Native PHY IP

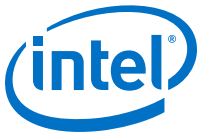
Figure 110. Configuring a PMA Parameter Using Native PHY IP Flow Chart



Note:
1. Refer to "PMA Adaptation" for PMA Adaptation Tab details.
2. Refer the "PMA Adaptation Options" table for details.
3. Refer to "PMA Bring Up Flow."
4. Refer to "PMA Parameters."
5. Refer to the "Configuring a PMA Parameter Using Native PHY IP" design example for details.

Related Information

- PMA Parameters on page 33
- PMA Adaptation Parameters on page 43



- PMA Bring Up Flow on page 83
- Loading a PMA Configuration on page 204

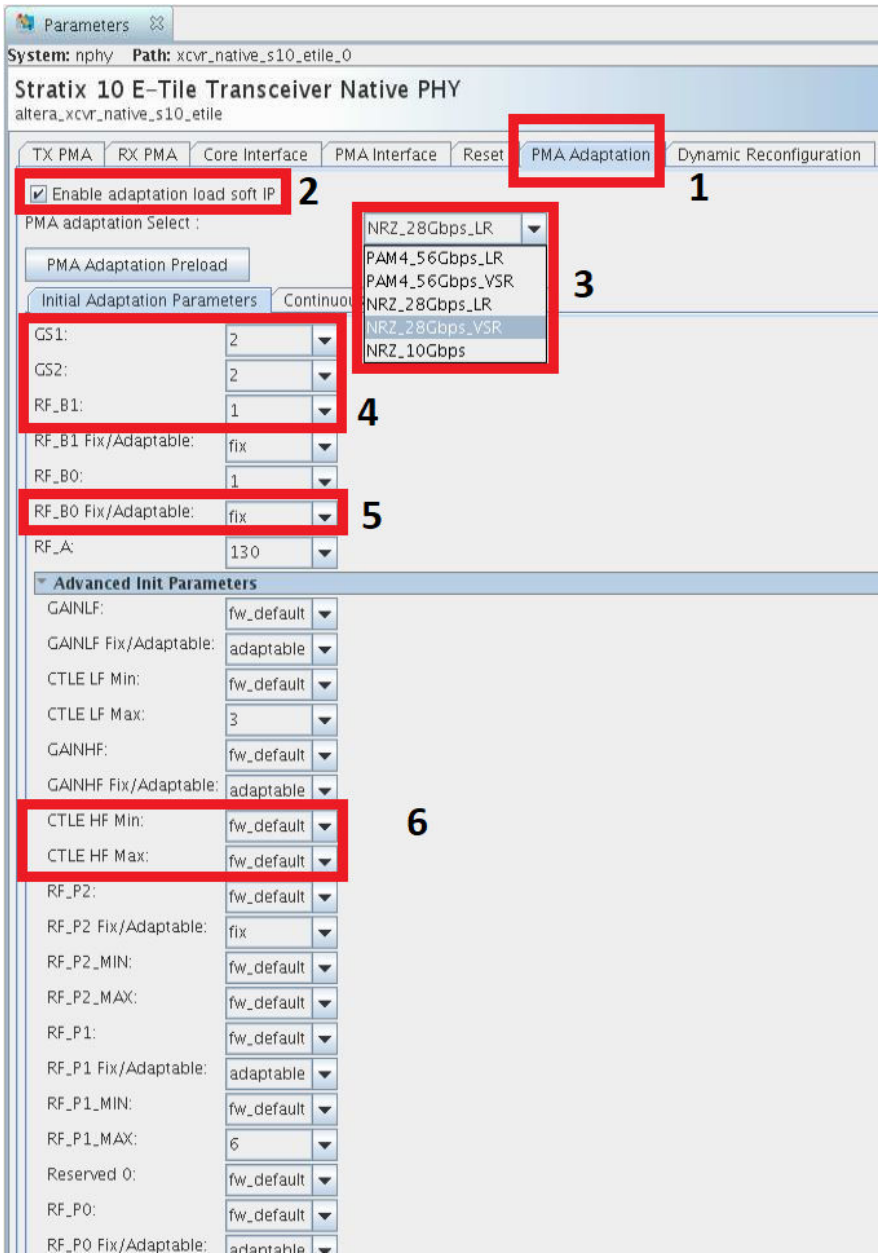
8.12.2. Native PHY IP GUI Details

Refer to *PMA Adaptation* for more details.

For PMA bring-up, refer to the DTF flow in *PMA Bring Up Flow*. To get optimal performance from the PMA across dynamic temperature conditions, PMA parameter tuning is required before initiating receiver initial adaptation and receiver continuous adaptation.

What follows is the Native PHY IP GUI configuration flow.

Figure 111. GUI for Initial Adaptation PMA Configuration Setup



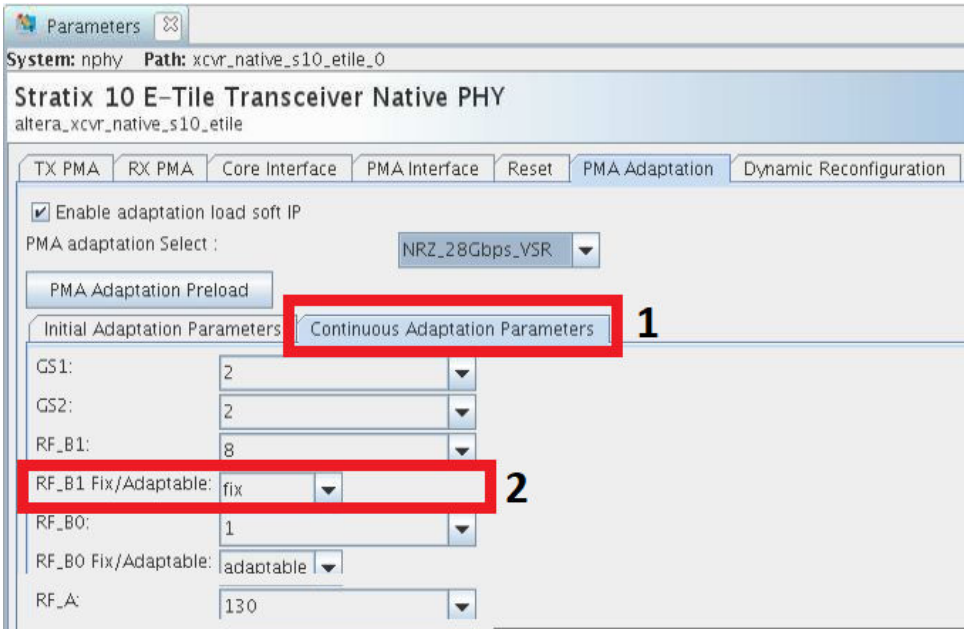
1. The **PMA Adaptation** tab is used to configure the transceiver PMA parameters to compensate the channel loss profile.
2. To enable this IP feature, enable adaptation load soft IP.
3. You can select the PMA configuration that configures the PMA AFE parameters to the required settings before initiating initial adaptation and continuous adaptation. The PMA configurations listed have been validated across PVT as per IEEE 802.3bs/bj specifications. If you have a different test setup, you must tune some of the parameters to achieve the optimal performance across the PVT.
4. Initial adaptation and continuous adaptation PMA parameter options are:

- `fw_default`
- all legal values

5. Initial Adaptation and continuous adaptation parameter adaptable options are:
   - fix
   - adaptable

   Some parameters are adaptable in order to compensate for loss. They may also be fixed which means that the parameter does not adapt when initial adaptation or continuous adaptation runs.

Figure 112. GUI for Continuous Adaptation PMA Configuration Setup



1. Continuous Adaptation parameter adaptable: options are fix and adaptable
2. Continuous Adaptation parameter values: options are `same_as_initial_parameter` and all legal values

Multi-PMA-Configuration Support

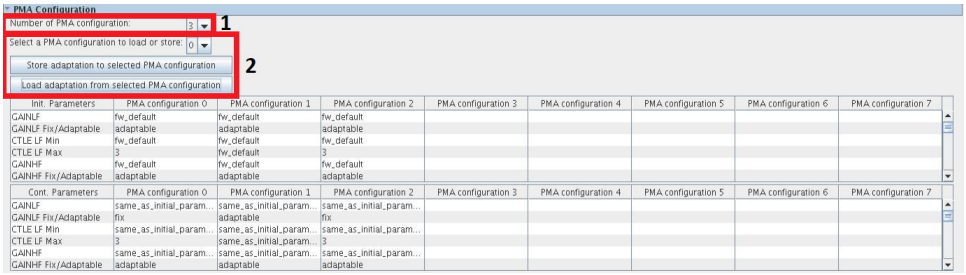
One Native PHY IP core supports up to eight different PMA configurations. You can choose one PMA configuration and load it for editing. The table shows the settings in each PMA configuration, which provides an overview of all PMA configurations.

PMA Adaptation settings can be further tuned for any channel. Refer to *PMA Registers 0x200 to 0x203 Usage*.

You can select the soft registers to choose and load the test configuration. The status of the test configuration load feature can be monitored using `rcp_load_finish`.

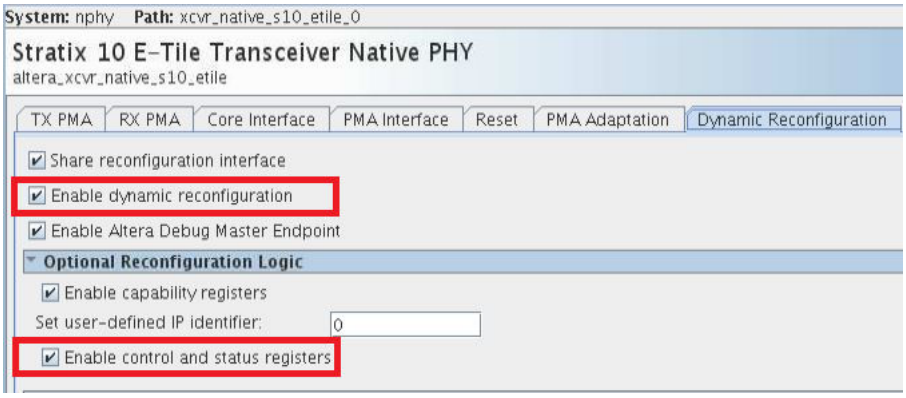
Figure 113. Multi-PMA-Configuration Settings

(32)



1. The IP supports eight PMA configurations (you can select them).
2. Load your PMA configuration, and apply it to all channels.

Figure 114. Enable Soft IP GUI



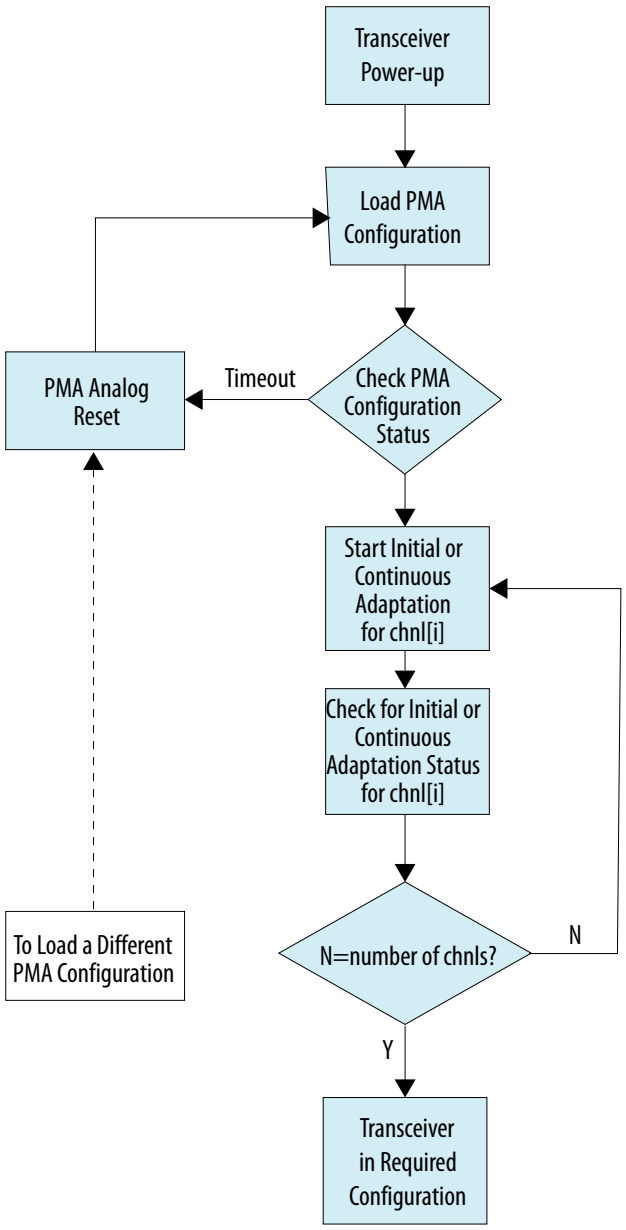
To enable the soft IP, turn on **Enable dynamic reconfiguration** and **Enable control and status registers**.

(32) When using PMA Adaptation if only one PMA configuration is used and that PMA configuration consists of all FW defaults (such as 28G NRZ VSR), a warning like this appears during the design compilation process:

```
Warning(16788): Net "rom[0][31]" does not have a driver at
alt_xcvr_native_rcp_load_rom_3gasjga.sv(44)
```

Either ignore this warning or, if there is only a PMA configuration with FW defaults, do not use PMA Adaptation.

Figure 115. Adaptation Working Flow



In the above flow, the transceiver reset is required only for non-Hard PRBS designs, meaning that the data is coming to the transceiver tile from the FPGA core.

Refer to *Loading a PMA Configuration* for more details.

Related Information

- PMA Adaptation Parameters on page 43
- PMA Bring Up Flow on page 83
- PMA Analog Reset on page 132

- Loading a PMA Configuration on page 204
- PMA Registers 0x200 to 0x203 Usage on page 235

8.12.3. Loading a PMA Configuration

This is an example of a single channel loading a single PMA configuration.

Set the operation mode (loopback mode and PRBS), setting internal serial loopback and PRBS31.

1. Write 0x200 = 0x0D. This selects internal serial loopback and PRBS31.
2. Write 0x201 = 0x00.
3. Write 0x202 = 0x00.
4. Write 0x203 = 0x93. This picks the opcode for SET_OPERATION_MODE.
5. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.

Load the PMA configuration using soft registers.

6. Write 0x40143 = 0x80. Loads the PMA configuration to all channels (0x40143 and 0x40144 can only be accessed from Channel 0).
7. Read 0x40144[0] until it reports 0x1. This ensures that this process did not time out.
8. Refer to the "Loading PMA Configuration Register START_ADAPTATION" figure for details.

Ensure that the PMA configuration is loaded to all channel PMA registers, initial adaptation, continuous adaptation, or both are run, loopback mode is set, PRBS is set, etc.

9. Write 0x200 = 0xD2. This is running initial adaptation only (would run continuous adaptation if selected above) loads the PMA configuration to the registers for this channel and again ensures PRBS31 is used for this command. To run continuous adaptation, refer to *PMA Registers 0x200 to 0x203 Usage*.
10. Write 0x201 = 0x02. This sets internal serial loopback again as part of this command.
11. Write 0x202 = 0x01. This enables initial adaptation (not running continuous adaptation in this case). For details on what can be done during continuous adaptation, refer to *PMA Bring Up Flow*.
12. Write 0x203 = 0x96. This picks the opcode for START_ADAPTATION.
13. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.
14. Write 0x200 = 0xF6. This is running initial adaptation and continuous adaptation, loads the PMA configuration to the registers for this channel, and disables PRBS to set mission mode.
15. Write 0x201 = 0x01. This sets external loopback and the 1st bit is part of disabling PRBS to set mission mode.
16. Write 0x202 = 0x03. This enables initial adaptation and continuous adaptation.
17. Write 0x203 = 0x96. This picks the opcode for START_ADAPTATION.
18. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.

Related Information

- PMA Bring Up Flow on page 83
- PMA Registers 0x200 to 0x203 Usage on page 235
  See the "Loading PMA Configuration Register START_ADAPTATION" figure.

8.13. Enabling Low Power Mode for Multiple Channels

For a given design instance, there is a set of base addresses that correspond to the relative channel for each.

NRZ addresses go one channel at a time (eight channels in this example).

Channel Number	Base Address
0	0x00000
1	0x80000
2	0x100000
3	0x180000
4	0x200000
5	0x280000
6	0x300000
7	0x380000

PAM4 addresses go in multiples of two since they use every other channel (four channels in this example).

Channel Number	Base Address
0	0x00000
1	0x100000
2	0x200000
3	0x300000

Because LPM is enabled and disabled on a per channel basis, each channel is addressed individually.

For NRZ Channels

To enable LPM for channel 0 use these register writes targeting base address 0x00000:

1. Write 0x01 to 0x200 to enable LPM, or write 0x00 to 0x200 to disable LPM.
2. Write 0x00 to 0x201 as this should be all 0's.
3. Write 0x00 to 0x202 to target the calling channel (channel 0 in this case).
4. Write 0x98 to 0x203 to use the OPCODE for LOW_POWER_MODE.
5. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.

To enable LPM for channel 5 use these register writes targeting base address 0x280000:

1. Write 0x01 to 0x200 to enable LPM, or write 0x00 to 0x200 to disable LPM.
2. Write 0x00 to 0x201 as this should be all 0's.
3. Write 0x00 to 0x202 to target the calling channel (channel 5 in this case).
4. Write 0x98 to 0x203 to use the OPCODE for LOW_POWER_MODE.
5. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.

For PAM4 Channels

To enable LPM for channel 0 use these register writes targeting base address 0x00000:

1. Write 0x01 to 0x200 to enable LPM, or write 0x00 to 0x200 to disable LPM.
2. Write 0x00 to 0x201 as this should be all 0's.
3. Write 0x00 to 0x202 to target the calling channel (channel 0 in this case).
4. Write 0x98 to 0x203 to use the OPCODE for LOW_POWER_MODE.
5. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.

To enable LPM for channel 3 use these register writes targeting base address 0x300000:

1. Write 0x01 to 0x200 to enable LPM, or write 0x00 to 0x200 to disable LPM.
2. Write 0x00 to 0x201 as this should be all 0's.
3. Write 0x00 to 0x202 to target the calling channel (channel 3 in this case).
4. Write 0x98 to 0x203 to use the OPCODE for LOW_POWER_MODE.
5. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.

8.14. Initializing an RX

For a given instance and channel number, do the following.

1. Set PLL the recalibration flag by using PMA attribute code 0x0011.
   a. Write 0x84[0] = 1'b1 to enable TX_PLL_RECAL flag.
   b. Write 0x84[1] = 1'b1 to enable RX_PLL_RECAL flag.
   c. Write 0x84[7:2] = 5'h00.
   d. Write 0x85[7:0] = 8'h00.
2. Disable the RX and leave the TX enabled by using PMA attribute code 0x0001.
   a. Write 0x84[0] = 1'b1 to enable TX.
   b. Write 0x84[1] = 1'b0 to disable RX.
   c. Write 0x84[2] = 1'b1 to enable TX output.
   d. Write 0x84[7:3] = 5'h00.
   e. Write 0x85[7:0] = 8'h00.
3. Set the RX channel divide by ratio by using PMA attribute code 0x0006.

a. Write 0x84[7:0] to set the data rate to the reference clock frequency ratio. See *Supported Data Rate Ratios for PMA Attribute Codes 0x0005 and 0x0006*.
b. Set the RX running rate.
    i. Write 0x85[2:0] = 3'b000 to set the RX running at more than 15 Gbaud per second.
    ii. Write 0x85[2:0] = 3'b001 to set the RX running at half rate.
    iii. Write 0x85[2:0] = 3'b010 to set the RX running at quarter rate.
    iv. Write 0x85[2:0] = 3'b011 to set the RX running at one-eighth rate.
c. Write 0x85[3] = 3'h0.
d. Select the RX reference clock
    i. Write 0x85[6] = 1'b0 to select `refclk_in_a` as the RX reference clock.
    ii. Write 0x85[6] = 1'b1 to select `refclk_in_b` as the RX reference clock.
e. Write 0x85[7] = 1'b1 to apply settings to both TX and RX.

4. Wait 20 milliseconds.
5. Set the TX/RX width mode by using PMA attribute code 0x0014.
    a. Set the TX width.
        i. Write 0x84[2:0] = 3'b001 to set the TX in 20-bit width mode (NRZ only).
        ii. Write 0x84[2:0] = 3'b011 to set the TX in 40-bit width mode (NRZ or PAM4).
        iii. Write 0x84[2:0] = 3'b100 to set the TX in 16-bit width mode (NRZ only).
        iv. Write 0x84[2:0] = 3'b101 to set the TX in 32-bit width mode (NRZ or PAM4).
        v. Write 0x84[2:0] = 3'b110 to set the TX in 64-bit width mode (PAM4 only).
    b. Set the TX mode.
        i. Write 0x84[3] = 1'b1 to set the TX in PAM4 mode.
        ii. Write 0x84[3] = 1'b0 to set the TX in NRZ mode.
    c. Set the RX width.
        i. Write 0x84[6:4] = 3'b001 to set the RX in 20-bit width mode (NRZ only).
        ii. Write 0x84[6:4] = 3'b011 to set the RX in 40-bit width mode (NRZ or PAM4).
        iii. Write 0x84[6:4] = 3'b100 to set the RX in 16-bit width mode (NRZ only).
        iv. Write 0x84[6:4] = 3'b101 to set the RX in 32-bit width mode (NRZ or PAM4).
        v. Write 0x84[6:4] = 3'b110 to set the RX in 64-bit width mode (PAM4 only).
    d. Set the RX mode.
        i. Write 0x84[7] = 1'b1 to set the RX in PAM4 mode.
        ii. Write 0x84[7] = 1'b0 to set the RX in NRZ mode.
    e. Write 0x85[7:0] = 8'h00.
6. Wait 1000 milliseconds.
7. Set the PMA PRBS settings by using PMA attribute code 0x0002.

a. Select the pattern.
   i. Write 0x84[2:0] = 3'b000 to set to PRBS7.
   ii. Write 0x84[2:0] = 3'b001 to set to PRBS9.
   iii. Write 0x84[2:0] = 3'b010 to set to PRBS11.
   iv. Write 0x84[2:0] = 3'b011 to set to PRBS15.
   v. Write 0x84[2:0] = 3'b100 to set to PRBS23.
   vi. Write 0x84[2:0] = 3'b101 to set to PRBS31.
   vii. Write 0x84[2:0] = 3'b110 to set to PRBS13.
   viii. Write 0x84[2:0] = 3'b111 to set to user-defined pattern. This disables the PRBS.
b. Write 0x84[3] = 1'b0.
c. Write 0x84[4] = 1'b1 to re-seed on error.
d. Write 0x84[5] = 1'b1 to auto-seed correct (generator goes from all 0s to all 1s).
e. Write 0x84[6] = 1'b0.
f. Write 0x84[7] = 1'b1 to stop on error (RX only).
g. Write 0x85[0] = 1'b1 to load the TX PRBS generator.
h. Write 0x85[1] = 1'b1 to enable the RX PRBS generator.
i. Write 0x85[7:2] = 6'h00.

8. Enable the RX and leave the TX enabled by using PMA attribute code 0x0001.
   a. Write 0x84[0] = 1'b1 to enable the TX.
   b. Write 0x84[1] = 1'b1 to enable the RX.
   c. Write 0x84[2] = 1'b1 to enable TX output.
   d. Write 0x84[7:3] = 5'h00.
   e. Write 0x85[7:0] = 8'h00.
9. Wait 1000 milliseconds.
10. Enable the electrical idle detector by using PMA attribute code 0x0020.
    a. Write 0x84[4:0] = 5'h00.
    b. Write 0x84[5] = 1'b1 to enable the Rx idle detector.
    c. Write 0x84[6] = 1'b0 to not update the idle detect threshold value.
    d. Write 0x84[7] = 1'b0.
    e. Write 0x85[7:0] = 8'h00.
11. Read the TX and RX states by using PMA attribute code 0x4026.
    a. Write 0x84[7:0] = 8'h00.
    b. Write 0x85[7:0] = 8'h00.
    c. Read 0x88[0]. 1'b1 indicates `tx_ready`.
    d. Read 0x88[1]. 1'b1 indicates `rx_ready`.
12. Read the RX fine lock by using PMA attribute code 0x401C.

a. Write 0x84[7:0] = 8'h00.
b. Write 0x85[7:0] = 8'h00.
c. Read 0x89[7]. 1'b1 indicates an RX fine lock.

Related Information

Supported Data Rate Ratios for PMA Attribute Codes 0x0005 and 0x0006 on page 239

8.15. Resetting the RX Equalization

1. Reset the receiver (RX) tuning by using PMA attribute code 0x000A.
   a. Write 0x84[7:0] = 8'hFF.
   b. Write 0x85[7:0] = 8'hFF.
2. Reset RF_P2, RF_P1, and RF_P0 to default values.
   - Use attribute code 0x002C with {0x85[7:0], 0x84[7:0]} = 0xD00.
   - Use attribute code 0x006C with {0x85[7:0], 0x84[7:0]} = 0x0.
   - Use attribute code 0x002C with {0x85[7:0], 0x84[7:0]} = 0xD01.
   - Use attribute code 0x006C with {0x85[7:0], 0x84[7:0]} = 0x0.
   - Use attribute code 0x002C with {0x85[7:0], 0x84[7:0]} = 0xD02.
   - Use attribute code 0x006C with {0x85[7:0], 0x84[7:0]} = 0x0.

   Refer to *Updating PMA Analog Parameters* for more details on using attribute codes 0x002C and 0x006C.

Related Information

Updating PMA Analog Parameters on page 232

8.16. Dynamic Reconfiguration Examples Revision History

Document Version	Changes
2020.01.31	Made the following changes: • Removed step 1 (setting the receiver termination to floating) in the *PMA Receiver Equalization Adaptation Usage Model*'s Initial Adaptation because it is not required. • Updated the *Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode* flow. • Added *Initializing an RX*. • Added *Resetting RX Equalization*.
2019.10.11	Made the following change: • Changed the last step of *Configuring the Attenuation Value (VOD)* from '1' to '0.'
2019.07.29	Made the following changes: • Added the RX phase slip step to *Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode*. • Added the PRBS31 steps to *Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode*. • Added *Enabling Low Power Mode for Multiple Channels*.
2019.04.19	Made the following change:

continued...

Document Version	Changes
	• In *Native PHY IP GUI Details*, added information about the warning that appears when using PMA Adaptation if only one PMA configuration is used and that PMA configuration consists of all FW defaults.
2019.02.04	Made the following changes: • Added *Configuring a PMA Parameter Using Native PHY IP*. • Added a step to *Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode*: Deassert `tx_reset/rx_reset`.
2018.10.08	Made the following changes: • Changed the steps in the "Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode" section. • Changed the bit range in the "PMA Attribute Code Error Inject Bits" table. • Added Step 9 to the "Configuring the Attenuation Value" section. • Added Step 9 and Step 18 to the "Configuring the Post Emphasis Value" section. • Added the "Example Use Case to Configure PMA Parameters" section. • Removed the "Reading 32-bit Errors" section. • Removed one-time adaptation from *PMA Receiver Equalization Adaptation Usage Model*. • For *Configuring the Post Emphasis Value*, added the option to set it to -2 (decimal) and 0xFE (hexadecimal). • Updated step #7 (resetting the internal controller) in *Reconfiguring the Duple PMA Using the Reset Controller in Automatic Mode*. • Added *Configuring a PMA Parameter Tunable by the Adaptive Engine*.
2018.07.18	Made the following changes: • Added three new topics: *Reading 32-bit Errors*, *Configuring TX Polarity for the PMA Driver*, and *Configuring RX Polarity for the PMA Driver*. • Switched the order of all write commands and added the step to "Write 0x8A[7] to 1 to clear the 0x8A[7] flag" in *User-Defined Pattern Example*. • Added *Configuring the Attenuation Value (VOD)*. • Added *Configuring the Post Emphasis Value*. • Added *Configuring `pretap1` Values*. • Changed the names of the adaptive equalization modes in the *PMA Receiver Equalization Adaptation Usage Model* section. • Added sub-step to step 8 in the *User-Defined Pattern Example* section. • Updated the sub-steps to step 10 in the *PRBS Usage Model* section.
2018.05.15	Made the following changes: • Added the "User-Defined Pattern Example" section. • Added a brief description and an example reconfiguration of a duplex PMA in the "Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode" section. • Created and added "Reconfiguring the Duplex PMA Using the Reset Controller in Automatic Mode." • Moved "PRBS Usage Model" and "PMA Receiver Equalization Adaptation Usage Model" to the "Dynamic Reconfiguration Usage Examples" chapter. • Added the "PMA Receiver Equalization Adaptation Flowchart." • Added the "Equalizer Bits" figure. • Added the "PRBS Bits" figure. • Added error injection instructions to the "PRBS Usage Model" section.
2018.01.31	Initial release.

9. Register Map

9.1. PMA Register Map

Refer to *Dynamic Reconfiguration Examples* for PMA register map usage examples.

The AVMM register space is not fully populated. Accessing undocumented registers results in the AVMM interface freezing.

Related Information

- Resetting Transceiver Channels on page 128
- E-Tile Transceiver PHY Architecture on page 69
- Dynamic Reconfiguration Examples on page 179

9.1.1. PMA Capability Registers

Table 71. PMA Capability Register Map

Name	Address	Bit Offset	Type	Description
IP Identifier	0x40000	[7:0]	read-only	ID 0: ID 0 to ID3 form a unique identifier for a given system.
	0x40001	[7:0]		ID 1
	0x40002	[7:0]		ID 2
	0x40003	[7:0]		ID 3
Status Register Enabled	0x40004	[0]	read-only	Indicates whether or not the status registers are enabled. 1'b1 indicates the feature is enabled.
Control Register Enabled	0x40005	[0]	read-only	Indicates whether or not the control registers are enabled. 1'b1 indicates the feature is enabled.
Number of channels	0x40010	[7:0]	read-only	Total Number of Channels in the Native PHY instance
Channel Number	0x40011	[7:0]	read-only	Channel Identifier: Unique channel ID for each channel in the instance
Duplex	0x40012	[1:0]	read-only	Shows transceiver mode: • 2'b11: Duplex

Refer to *PMA Attribute Codes* for details about the PMA attribute codes and values.

Related Information

PMA Attribute Codes on page 216

9.1.2. PMA Control and Status Registers

Table 72. PMA Control and Status Register Map

Name	Address	Bit Offset	Type	Description
RX Locked to Data Status	0x40080	[0]	read-only	Check the status of rx_is_lockedtodata
tx_ready Status	0x40081	[0]	read-only	Check the status of TX ready from the reset controller
rx_ready Status		[1]	read-only	Check the status of RX ready from the reset controller
tx_transfer_ready status	0x40082	[0]	read-only	Check the status of tx_transfer_ready
rx_transfer_ready status		[1]	read-only	Check the status of rx_transfer_ready
RX PMA Interface Reset	0x400E2	[0]	read-write	Set the status of rx_pmaif_reset Refer to *Resetting Transceiver Channels* for more information.
RX EMIB Reset		[1]	read-write	Set the status of rx_aib_reset Refer to *Resetting Transceiver Channels* for more information.
TX PMA Interface Reset		[2]	read-write	Set the status of tx_pmaif_reset Refer to *Resetting Transceiver Channels* for more information.
TX EMIB Reset		[3]	read-write	Set the status of tx_aib_reset Refer to *Resetting Transceiver Channels* for more information.
RX PMA Interface Reset Override		[4]	read-write	Override selects between soft AVMM and reset controller. Set to 0 for the reset controller to set the reset and 1 for AVMM register 0x400E2[0] to set the value.
RX EMIB Reset Override		[5]	read-write	Override selects between soft AVMM and reset controller. Set to 0 for the reset controller to set the reset and 1 for AVMM register 0x400E2[1] to set the value.
TX PMA Interface Reset Override		[6]	read-write	Override selects between soft AVMM and reset controller. Set to 0 for the reset controller to set the reset and 1 for AVMM register 0x400E2[2] to set the value.
TX EMIB Reset Override		[7]	read-write	Override selects between soft AVMM and reset controller. Set to 0 for the reset controller to set the reset and 1 for AVMM register 0x400E2[3] to set the value.
Configuration Profile Select	0x40140	[2:0]	read-write	Set the configuration profile the embedded streamer
Start Streaming		[7]	read-write	Set to 1'b1 to start streaming the new configuration profile
Busy Status Bit	0x40141	[0]	read-only	Bit is set to: • 1'b1: streaming is in progress • 1'b0: streaming is complete
PMA Configuration	0x40143	[7]	read-write	Request PMA configuration load

continued...

Name	Address	Bit Offset	Type	Description
Control		[2:0]	read-write	Select PMA configuration
PMA Configuration Loading Status	0x40144	[0]	read-only	rcp_load_finish
		[1]	read-only	rcp_load_timeout
		[2]	read-only	rcp_load_busy

Registers 0x40143 and 0x40144 are located only in Channel 0's address space in a multi-channel Native PHY IP instance.

Refer to *PMA Attribute Codes* for details about the PMA attribute codes and values.

Refer to *Multiple Reconfiguration Profiles* for more information about using the embedded streamer.

Related Information

- PMA Attribute Codes on page 216
- Multiple Reconfiguration Profiles on page 154

9.1.3. PMA AVMM Registers

Table 73. PMA AVMM Registers

Address	Bit Offset	Description
0x4	[0]	TX datapath clock enable
	[1]	Transmit full clock out (PMA Clock) enable
	[4:2]	Transmit data-input select
	[5]	Transmit full clock out (clk_tx_adapt) select
	[6]	Transmit clock datapath select
	[7]	Transmit adaptation order select. Determines how 64 bits are sent to 32-bit transceiver channel
0x5	[1:0]	Transmit multi-lane data select
	[2]	TX Gearbox clock enable
	[3]	TX datapath clock enable
	[4]	TX PCS div2 clock input enable
	[5]	TX FEC div2 clock input enable
	[6]	TX EHIP div2 clock input enable
	[7]	TX direct clock input enable
0x6	[0]	RX datapath clock enable
	[1]	Receive full clock out (rx_pma_clk) enable
	[2]	Receive half clock out (rx_pcs_clk) enable

continued...

Address	Bit Offset	Description
	[3]	Receive div66 clock out (`rx_pcs_div66_clk`) enable
	[4]	Receiver adaptation order select. Determines how 64 bits are combined from 32-bit transceiver channel
	[6:5]	Receiver adapter data select
	[7]	Receiver reverse bit order in Gearbox
0x7	[0]	Receiver reverse 64/66 sync header bit order in Gearbox
	[1]	RX FIFO Read clock enable
	[2]	Receive Gearbox and FIFO write clock enable
	[4:3]	Receive direct-data mode multi-lane data select. Only active if `cfg_rx_adapter_sel` is not equal to b'01. These are one-hot encoded
	[6:5]	Select RX FIFO Read clock
	[7]	RX adapter clock enable
0x8	[0]	Reverse data bit transmission order in TX Gearbox
	[1]	Reverse 64/66 sync header bit order transmission in TX Gearbox
	[3]	Dynamic bitslip enable for TX Gearbox
	[5]	Specify 64/66 sync header location in TX Gearbox
0x9	[1:0]	TX Deskew multi-lane mode select
	[3:2]	TX deskew bits 00 = not yet received a deskew-bit 01 = not aligned 10 = received 1 set of aligned deskew-bits 11 = received 16 sets of aligned deskew-bits
	[4]	TX deskew alignment status 0 = not aligned or not enabled or didn't receive a deskew-bit 1 = aligned
	[5]	RX FIFO bit-67 select
0xA	[2:0]	Transmit deskew enable (using one-hot encoding)
	[5]	Dynamic rx_bitslip enable
0x10	[4:0]	Transceiver interface RX FIFO empty threshold
	[7:6]	Transceiver interface RX FIFO almost empty threshold
0x11	[2:0]	Transceiver interface RX FIFO almost empty threshold
	[7:4]	Transceiver interface RX FIFO full threshold
0x12	[0]	Transceiver interface RX FIFO full threshold
	[6:2]	Transceiver interface RX FIFO almost full threshold
0x13	[6]	RX FIFO Read when Empty
	[7]	RX FIFO Write when Full
0x14	[4:0]	Transceiver interface TX FIFO empty threshold
	[7:6]	Transceiver interface TX FIFO almost empty threshold
0x15	[2:0]	Transceiver interface TX FIFO almost empty threshold

continued...

Address	Bit Offset	Description
	[7:4]	Transceiver interface TX FIFO full threshold
0x16	[0]	Transceiver interface TX FIFO full threshold
	[6:2]	Transceiver interface TX FIFO almost full threshold
0x17	[5:4]	TX FIFO Phase Compensation mode
	[6]	TX FIFO Write when Full
	[7]	TX FIFO Read when Empty
0x1C	[7:0]	Transmit output value [31:0] when the `user_reset` is active (after FPGA initialization)
0x1D	[7:0]	Transmit output value [31:0] when the `user_reset` is active (after FPGA initialization)
0x1E	[7:0]	Transmit output value [31:0] when the `user_reset` is active (after FPGA initialization)
0x1F	[7:0]	Transmit output value [31:0] when the `user_reset` is active (after FPGA initialization)
0x20	[7:0]	Transmit output value [63:32] when the `user_reset` is active (after FPGA initialization)
0x21	[7:0]	Transmit output value [63:32] when the `user_reset` is active (after FPGA initialization)
0x22	[7:0]	Transmit output value [63:32] when the `user_reset` is active (after FPGA initialization)
0x23	[7:0]	Transmit output value [63:32] when the `user_reset` is active (after FPGA initialization)
0x24	[2:0]	Transmit output value [66:64] when the `user_reset` is active (after FPGA initialization)
0x34	[1:0]	Serialization factor for `rx_bit_counter`
	[7:4]	The value at which `rx_bit_counter` should reset to 0. Set to 5280-32 for RS-FEC
0x35	[7:0]	The value at which `rx_bit_counter` should reset to 0. Set to 5280-32 for RS-FEC
0x36	[0]	The value at which `rx_bit_counter` should reset to 0. Set to 5280-32 for RS-FEC
	[3]	Read-Write self clear
	[4]	transmit div66 clock out (`tx_pcs_div66_clk`) enable
0x37	[0]	Transmit `sclk_enable`
	[2:1]	Increment TX FIFO latency select
	[4]	Receive sclk_enable
	[6:5]	Increment RX FIFO latency select
	[7]	Async latency pulse select
0x38	[0]	Duty cycle correction: duty cycle correction bypass disable
	[1]	DCC: DCC master enable
	[2]	DCC: select continuous cal
0x3C	[1]	DCC : enable for FSM
0x80	[7:0]	Core PMA attribute control
0x81	[7:0]	Core PMA attribute control
0x84	[7:0]	PMA attribute data
0x85	[7:0]	PMA attribute data
0x86	[7:0]	PMA attribute code

continued...

Address	Bit Offset	Description
0x87	[7:0]	PMA attribute code
0x88	[7:0]	Lower byte of the PMA attribute code return value
0x89	[7:0]	Upper byte of the PMA attribute code return value
0x8A	[7]	Indicates the PMA attribute has been transmitted to the PMA successfully
0x8B	[0]	1'b0 indicates the PMA has finished acting on the PMA attribute and the PMA attribute code return value is available on registers 0x88/0x89
0x90	[0]	Loads the contents of registers 0x84 to 0x87 (which form the PMA attribute contents) to the PMA
0x91	[0]	Loads either the initial PMA setting or the last selected profile into the PMA. Used when changing the PMA's reference clock as described in *Switching Reference Clocks*.
0x95	[5]	1'b1 calibrates the PMA when loading new settings
0xEC	[3:0]	Selects reference clocks [0-8] muxed onto `refclkin_in_A`
	[7:4]	Selects which reference clock [0-8] is mapped to `refclk4` in the Native PHY IP core
0xEE	[3:0]	Selects which reference clock [0-8] is mapped to `refclk0` in the Native PHY IP core
	[7:4]	Selects which reference clock [0-8] is mapped to `refclk1` in the Native PHY IP core
0xEF	[3:0]	Selects which reference clock [0-8] is mapped to `refclk2` in the Native PHY IP core
	[7:4]	Selects which reference clock [0-8] is mapped to `refclk3` in the Native PHY IP core
0x200	[7:0]	Places the PMA in analog reset or sets up the PMA operating mode (see *PMA Registers 0x200 to 0x203 Usage*)
0x201	[7:0]	Places the PMA in analog reset or sets up the PMA operating mode (see *PMA Registers 0x200 to 0x203 Usage*)
0x202	[7:0]	Places the PMA in analog reset or sets up the PMA operating mode (see *PMA Registers 0x200 to 0x203 Usage*)
0x203	[7:0]	Places the PMA in analog reset or sets up the PMA operating mode (see *PMA Registers 0x200 to 0x203 Usage*)
0x207	[0]	0 indicates the operation completed successfully
0x204	[7:0]	Returns the physical channel number in order to load the IP configuration to a different channel
0x207	[7]	1 indicates the last operation on registers 0x200 to 0x203 completed. You must also read 0x207[0] to check whether the operation was successful.

Related Information

- Switching Reference Clocks on page 161
- PMA Registers 0x200 to 0x203 Usage on page 235

9.2. PMA Attribute Codes

Use the following attribute codes to set registers 0x87[7:0] down to 0x84[7:0] in the PMA register map to send or receive attribute values to or from the PMA. Write the PMA attribute code values on registers {0x87,0x86}.

9.2.1. 0x0001: PMA Enable/Disable

Attribute Code

0x0001

Description

Turns the PMA on or off.

- 0x84[0]:
  - 1'b1 to enable TX
  - 1'b0 to disable TX
- 0x84[1]:
  - 1'b1 to enable RX
  - 1'b0 to disable RX
- 0x84[2]:
  - 1'b1 to enable TX output
  - 1'b0 to disable TX output
- 0x84[7:3]: 5'h00
- 0x85[7:0]: 8'h00

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x0001 indicates the PMA attribute has been updated.

9.2.2. 0x0002: PMA PRBS Settings

Attribute Code

0x0002

Description

Controls the PRBS settings for the PMA:

- 0x84[2:0]: 3'b000 to set to PRBS7
- 0x84[2:0]: 3'b001 to set to PRBS9
- 0x84[2:0]: 3'b010 to set to PRBS11
- 0x84[2:0]: 3'b011 to set to PRBS15
- 0x84[2:0]: 3'b100 to set to PRBS23
- 0x84[2:0]: 3'b101 to set to PRBS31
- 0x84[2:0]: 3'b110 to set to PRBS13
- 0x84[2:0]: 3’b111 to set to user-defined pattern. Setting it to user-defined pattern disables the PRBS.

- 0x84[3]: 1'b0
- 0x84[4]: 1'b1 to re-seed on error
- 0x84[5]: 1'b1 to auto-seed correct (generator goes from all 0s to all 1s)
- 0x84[6]: 1'b0
- 0x84[7]: 1'b1 to stop on error (RX only)
- 0x85[0]: 1'b1 to load TX PRBS Generator
- 0x85[1]: 1'b1 to enable RX PRBS Generator
- 0x85[7:2]: 6'h00

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x0000: Failed due to background processes. Wait for some time and re-issue.

0x0002: Success

Refer to the *PRBS Usage Model* section for more details.

Related Information

PRBS Usage Model on page 183

9.2.3. 0x0003: Data Comparison Set Up and Start/Stop

Attribute Code

0x0003

Description

Sets up and starts/stops data comparisons.

- 0x84[0]: 1'b1 to enable the data comparison based on the reset of the data bits values
- 0x84[1]: 1'b0 to compare the ratio of 1s/0s. 1'b1 to compare the sum
- 0x84[3:2]: 2'h0
- 0x84[6:4]: 3'b000 to select `rx_data` as source data for comparison
- 0x84[6:4]: 3'b010 to select RX pattern generator data as source data for comparison
- 0x84[6:4]: 3'b011 to select `tx_data` as source data for comparison
- 0x84[6:4]: 3'b100 to select `tx_prbs` as source data for comparison
- 0x84[6:4]: 3'b110 to select 20'h00000 as source data for comparison
- 0x84[6:4]: 3'b111 to select 20'hFFFFF as source data for comparison
- 0x84[7]: 1'b0
- 0x85[2:0]: 3'b000 to select `rx_data` as data to be compared against

- 0x85[2:0]: 3'b010 to select RX pattern generator data as data to be compared against
- 0x85[2:0]: 3'b011 to select `tx_data` as data to be compared against
- 0x85[2:0]: 3'b100 to select `tx_prbs` as data to be compared against
- 0x85[2:0]: 3'b110 to select 20'h00000 as data to be compared against
- 0x85[7:3]: 5'h00

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x0000: Failed due to background processes. Wait for some time and re-issue.

0x0003: Success

9.2.4. 0x0005: TX Channel Divide By Ratio

Attribute Code

0x0005

Description

Configures TX of transceiver channel to the appropriate divide-by ratio. See *Supported Data Rate Ratios for PMA Attribute Codes 0x0005 and 0x0006* for valid settings.

- 0x84[7:0] Set the data rate to reference clock frequency ratio
- 0x85[2:0]: 3'b000 to set the TX running at more than 15 Gbaud per second
- 0x85[2:0]: 3'b001 to set the TX running at half rate
- 0x85[2:0]: 3'b010 to set the TX running at quarter rate
- 0x85[2:0]: 3'b011 to set the TX running at one-eighth rate
- 0x85[3]: 1'b0
- 0x85[4]: 1'b1 to configure the PMA channel as a slave channel when multiple channels are bonded
- 0x85[5]: 1'b0
- 0x85[6]: 1'b0 to select `refclk_in_a` as the TX reference clock
- 0x85[6]: 1'b1 to select `refclk_in_b` as the TX reference clock
- 0x85[7]: 1'b1 to apply settings to both TX and RX

PMA Can Be Running While Updating PMA Attribute?

No

Return Value {0x89[7:0],0x88[7:0]}

0x00FF: Invalid configuration

0x0005: Success

Related Information

Supported Data Rate Ratios for PMA Attribute Codes 0x0005 and 0x0006 on page 239

9.2.5. 0x0006: RX Channel Divide By Ratio

Attribute Code

0x0006

Description

Configures RX of transceiver channel to the appropriate divide-by ratio. See *Supported Data Rate Ratios for PMA Attribute Codes 0x0005 and 0x0006* for valid settings.

- 0x84[7:0] Set the data rate to reference clock frequency ratio
- 0x85[2:0]: 3'b000 to set the RX running at more than 15 Gbaud per second
- 0x85[2:0]: 3'b001 to set the RX running at half rate
- 0x85[2:0]: 3'b010 to set the RX running at quarter rate
- 0x85[2:0]: 3'b011 to set the RX running at one-eighth rate
- 0x85[3]: 3'h0
- 0x85[6]: 1'b0 to select `refclk_in_a` as the RX reference clock
- 0x85[6]: 1'b1 to select `refclk_in_b` as the RX reference clock
- 0x85[7]: 1'b1 to apply settings to both TX and RX

PMA Can Be Running While Updating PMA Attribute?

No

Return Value {0x89[7:0],0x88[7:0]}

0x00FF: Invalid configuration

0x0006: Success

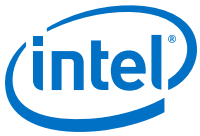
Related Information

Supported Data Rate Ratios for PMA Attribute Codes 0x0005 and 0x0006 on page 239

9.2.6. 0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control

Attribute Code

0x0008



Description

Controls turning on/off internal serial loopback or reverse parallel loopback.

- 0x84[0]: 1'b1 to select internal serial loopback. 1'b0 to disable internal serial loopback.
- 0x84[3:1]: 3'h0
- 0x84[4]: 1'b1 to select reverse parallel loopback. 1'b0 to disable reverse parallel loopback.
- 0x84[7:5]: 3'h0
- 0x85[0]: 1'b1 to change the internal serial loopback settings
- 0x85[1]: 1'b1 to set the reverse parallel loopback settings
- 0x85[7:2]: 6'h00

Only one loopback mode can be enabled at a time. For example, set 0x84[7:0] to 0x01 and 0x85[7:0] to 0x03 to enable internal serial loopback and disable reverse parallel loopback.

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x0008

9.2.7. 0x000A: Receiver Tuning Controls

Attribute Code

0x000A

Description

Launches receiver tuning, enables/disables adaptive receiver tuning, and controls how some tuning knobs are used.

- 0x84[3:0]: 4'b0001 to run initial adaptive equalization
- 0x84[3:0]: 4'b0010 to freeze continuous adaptive equalization to allow the error counter to be read out when the PRBS generator/verifier is enabled (the Freeze adaptation mode)
- 0x84[3:0]: 4'b0110 to run continuous adaptive equalization
- 0x84[7:4]: 0x0
- 0x85[7:0]: 0x00
- 0x84[7:0]: 8'hFF to reset the RX equalization
- 0x85[7:0]: 8'hFF to reset the RX equalization

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x000A

9.2.8. 0x000E: RX Phase Slip

Attribute Code

0x000E

Description

This attribute code is used during the dynamic reconfiguration rate change to adjust the phase offset of a recovered clock to the intended data rate selected based on your requirements.

- 0x84[7:0]: 0x00
- 0x85[5:0]: Phase slip count (each slip slips by 1 UI). Recommended phase slip settings based on the RX PMA modulation type (NRZ or PAM4) and the RX PMA interface width settings in the E-tile transceiver Native PHY IP core:
  - NRZ Mode
    - RX PMA interface width = 16: phase slip = 12
    - RX PMA interface width = 20: phase slip = 16
    - RX PMA interface width = 32: phase slip = 28
    - RX PMA interface width = 40: phase slip = 36
  - PAM4 Mode
    - RX PMA interface width = 32: phase slip = 12
    - RX PMA interface width = 40: phase slip = 16
    - RX PMA interface width = 64: phase slip = 28
- 0x85[6]: 1'b0
- 0x85[7]: 1'b1 always asserted to apply these slips every time RX is enabled

Disabling and enabling the PMA multiple times without a PMA analog reset results in multiple RX phase slips being applied. The RX phase slip is additive.

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

N/A

9.2.9. 0x0011: PMA TX/RX Calibration

Attribute Code

0x0011

Description

Sets whether the PMA transmitter path, receiver path, or both are calibrated the next time the PMA is enabled when using PMA attribute code 0x0001.

- 0x84[0]: 1'b1 to recalibrate the PMA transmitter path the next time the transmitter is enabled
- 0x84[1]: 1'b1 to recalibrate the PMA receiver path the next time the receiver is enabled
- 0x84[7:2]: 6'h00
- 8x85[7:0]: 8'h00

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x0011

9.2.10. 0x0013: TX/RX Polarity and Gray Code Encoding

Attribute Code

0x0013

Description

Changes the state of TX polarity, RX polarity, or both. Also changes the state of TX Gray code, TX Precode, and TX Swizzle, RX Gray code, RX Precode, and RX Swizzle or both groups of settings.

- 0x84[0]: 1'b1 to invert TX polarity
- 0x84[1]: 1'b1 to enable TX Gray code in PAM4 mode
- 0x84[2]: 1'b1 to enable TX Precode (1/1+D) in PAM4 mode
- 0x84[3]: 1'b0 for even bits to be mapped to PAM4 LSB (TX Swizzle)
- 0x84[3]: 1'b1 for even bits to be mapped to PAM4 MSB (TX Swizzle)
- 0x84[4]: 1'b1 to invert RX polarity
- 0x84[5]: 1'b1 to enable RX Gray code in PAM4 mode
- 0x84[6]: 1'b1 to enable RX Precode (1/1+D) in PAM4 mode
- 0x84[7]: 1'b0 for RX even bits to be mapped to PAM4 LSB (RX Swizzle)
- 0x84[7]: 1'b1 for RX even bits to be mapped to PAM4 MSB (RX Swizzle)
- 0x85[0]: 1'b1 to set TX polarity to provided value in 0x84[0]
- 0x85[1]: 1'b1 to set RX polarity to provided value in 0x84[4]
- 0x85[2]: 1'b1 to set TX PMA4 MSB/LSB mapping, gray code encoding, precoding to provided values on 0x84[3:1]
- 0x85[3]: 1'b1 to set RX PMA4 MSB/LSB mapping, gray code encoding, precoding to provided values on 0x84[7:4]
- 0x85[7:4]: 4'h0

PMA Can Be Running While Updating PMA Attribute?

No

Return Value {0x89[7:0],0x88[7:0]}

0x0013

9.2.11. 0x0014: TX/RX Width Mode

Attribute Code

0x0014

Description

Controls the TX/RX width mode.

- 0x84[2:0]: 3'b001 to set TX in 20-bit width (NRZ only)
- 0x84[2:0]: 3'b011 to set TX in 40-bit width (NRZ or PAM4)
- 0x84[2:0]: 3'b100 to set TX in 16-bit width (NRZ only)
- 0x84[2:0]: 3'b101 to set TX in 32-bit width (NRZ or PAM4)
- 0x84[2:0]: 3'b110 to set TX in 64-bit width (PAM4 only)
- 0x84[3]: 1'b1 to set TX in PAM4 mode
- 0x84[3]: 1'b0 to set TX in NRZ mode
- 0x84[6:4]: 3'b001 to set RX in 20-bit width (NRZ only)
- 0x84[6:4]: 3'b011 to set RX in 40-bit width (NRZ or PAM4)
- 0x84[6:4]: 3'b100 to set RX in 16-bit width (NRZ only)
- 0x84[6:4]: 3'b101 to set RX in 32-bit width (NRZ or PAM4)
- 0x84[6:4]: 3'b110 to set RX in 64-bit width (PAM4 only)
- 0x84[7]: 1'b1 to set RX in PAM4 mode
- 0x84[7]: 1'b0 to set RX in NRZ mode
- 0x85[7:0]: 8'h00

PMA Can Be Running While Updating PMA Attribute?

No

Return Value {0x89[7:0],0x88[7:0]}

0x0014

9.2.12. 0x0015: TX Equalization

Attribute Code

0x0015

Description

Loads the TX EQ with the provided values.

- 0x84[7:0]: Set the TX Pre-emphasis, Post-emphasis, attenuation values in 2's complement. Refer to the "TX Equalization Settings for PAM4 and NRZ Signals" table in *TX Equalizer* for the valid TX equalization range.
- 0x85[0]: 1'b0 to set the TX EQ with values in 0x84[7:0]
- 0x85[0]: 1'b1 to read the current EQ value. 0x84[7:0] are ignored
- 0x85[3:1]: 3'h0
- 0x85[7:4]: 4'b0000 to set/read the TX pre-emphasis tap 1 value
- 0x85[7:4]: 4'b0011 to set/read the TX pre-emphasis tap 3 value
- 0x85[7:4]: 4'b0100 to set/read the main tap value
- 0x85[7:4]: 4'b1000 to set/read the Post-emphasis tap 1 value
- 0x85[7:4]: 4'b1100 to set/read the TX pre-emphasis tap 2 value

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

If 0x85[0] is 1'b1: EQ value

If 0x85[0] is 1'b0:

- 0x0015 if success
- 0x0000 if failed to apply since new EQ setting exceeds limit of allowed EQ

Related Information

TX Equalizer on page 73

9.2.13. 0x0017: Error Counter Reset

Attribute Code

0x0017

Description

Resets the error counter.

- 0x84[7:0]: 8'h00
- 0x85[7:0]: 8'h00

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x0017 for successful error counter reset

9.2.14. 0x0018: Status/Debug Register

Attribute Code

0x0018

Description

Sets up which status/debug register to be used for subsequent read/write operations.

- {0x85[7:0],0x84[7:0]}:
  - 0x0000: Select 80b-wide TX data to be written 10b at a time starting from the LSB by asserting PMA attribute code 0x19 (assert the PMA attribute code eight times)
- {0x85[7:0],0x84[7:0]}:
  - 0x0001: Select 80b-wide RX data to be written 10b at a time starting from the LSB by asserting PMA attribute code 0x19 (assert the PMA attribute code eight times)
- {0x85[7:0],0x84[7:0]}:
  - 0x0002: Select 30b-wide error timer to be read 16b at a time starting from the LSB by asserting PMA attribute code 0x1A (assert the PMA attribute code two times)
- {0x85[7:0],0x84[7:0]}:
  - 0x0003: Select 32b-wide error counter to be read 16b at a time starting from the LSB by asserting PMA attribute code 0x1A (assert the PMA attribute code two times)
- {0x85[7:0],0x84[7:0]}:
  - 0x0004: Select 80b-wide recovered RX data to be read 10b at a time starting from the LSB by asserting PMA attribute code 0x19 (assert the PMA attribute code eight times)

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x0018

9.2.15. 0x0019: Status/Debug Register Next Write Field

Attribute Code

0x0019

Description

Writes the next field of a status/debug register. 0x85[7:0], 0x84[7:0] represent the value to be written.

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x0019

9.2.16. 0x001A: Status/Debug Register Next Read Field

Attribute Code

0x001A

Description

Reads the next field of a status/debug register.

0x84[7:0]: 8'h00

0x85[7:0] 8'h00

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

Varies

9.2.17. 0x001B: TX Error Injection Signal

Attribute Code

0x001B

Description

Switches the TX error injection signal on or off for the number of times requested.

- {0x85[7:0],0x84[7:0]}:
  - Number of errors to inject

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x001B

9.2.18. 0x001C: Incoming RX Data Capture

Attribute Code

0x001C

Description

Captures incoming RX data to be read by the status/debug register read

- 0x84[7:0]: 8'h00
- 0x85[7:0]: 8'h00

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x001C

9.2.19. 0x001E: Error Count Status

Attribute Code

0x001E

Description

Reads the error count status register

- 0x84[7:0]: 8'h00
- 0x85[7:0]: 8'h00

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x88[4]: 1'b1 if error occurred

0x88[4]: 1'b0 if no error

9.2.20. 0x0020: Electrical Idle Detector

Attribute Code

0x0020

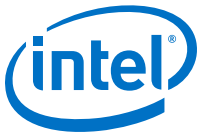
Description

Configures the electrical idle detector.

- 0x84[5]: 1'b1 Enable RX idle detector
- 0x84[6]: 1'b1 Update the idle detect threshold with values in 0x85[3:0]. If set, 0x84[5:0] is ignored.
- 0x85[3:0]: 8'h00 Idle detect threshold value. Only updated if 0x84[6] is set.

PMA Can Be Running While Updating PMA Attribute?

Yes



Return Value {0x89[7:0],0x88[7:0]}

0x0020

9.2.21. 0x002B: RX Termination and TX Driver Tri-state Behavior

Attribute Code

0x002B

Description

Sets RX termination and TX driver tri-state behavior.

- 0x84[0]: 1'b1: RX termination to VCCH_GXE
- 0x84[1]: 1'b0: Active termination to ground/VCCH_GXE based on 0x84[0]
- 0x84[1]: 1'b1: Termination undriven (floating)
- 0x84[3:2]: 2'h0
- 0x84[4]: 1'b0: Normal TX behavior (both TX outputs driven to VCCH_GXE) (default)
- 0x84[4]: 1'b1: TX output tristated when disabled
- 0x84[5]: 1'b0: Configure RX termination (0x84[4] is ignored)
- 0x84[5]: 1'b1: Configure TX tri-state (0x84[1:0] are ignored)
- 0x84[7:6]: 2'h0
- 0x85[7:0]: 8'h00

When you use external AC-coupling capacitors, set RX termination to VCCH_GXE; whereas, when you do not use external AC-coupling caps, set RX termination to floating. The RX termination is floating on power-up and after a PMA analog reset.

Table 74. Configuring TX Tri-state

{0x89[7:0], 0x88[7:0]}	{0x85[7:0],0x84[7:0]}	Description
0x002B	0x0001	RX termination set to VCCH_GXE - TX tri-state behavior unchanged
0x002B	0x0002	RX termination set to float - TX tri-state behavior unchanged
0x002B	0x0020	Disable tri-state (default) - RX termination setting unchanged
0x002B	0x0030	Enable TX tristate - RX termination setting unchanged

PMA Can Be Running While Updating PMA Attribute?

No

Return Value {0x89[7:0],0x88[7:0]}

0x002B

9.2.22. 0x0030: PMA Mux Clock Swap

Attribute Code

0x0030

Description

Switch between `refclk_in_A` and `reflck_in_B` on the PMA mux to do dynamic reconfiguration on the reference clocks.

- 0x84[2:0]: 3'b000 to set to `refclk_in_A` (default)
- 0x84[2:0]: 3'b011 to set to `refclk_in_B`
- 0x84[7:3]: 5b'00000
- 0x85[7:0]: 0x00

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x0030

9.2.23. 0x0126: Read Receiver Tuning Parameters

Attribute Code

0x0126

Description

Read receiver tuning parameters and settings.

- 0x84[7:0]: 8'h00
- 0x85[7:0]: 8'h0B

PMA Can Be Running While Updating PMA Attribute?

Yes

Return Value {0x89[7:0],0x88[7:0]}

0x88[0]: 1'b1 if initial coarse adaptive equalization in progress

0x88[4]: 1'b1 if initial coarse adaptive equalization is enabled

0x88[5]: 1'b1 if fine adaptive equalization is enabled

0x88[6]: 1'b1 if continuous adaptive equalization is enabled

0x88[7]: 1'b1 if input offset correction finished

9.2.24. Reading and Writing PMA Analog Parameters Using Attributes

The PMA analog parameters listed in Table 39 on page 78 can be read and set using PMA attributes. The analog parameters can be read or changed when the transceiver is running.

Related Information

- Configuring a PMA Parameter Tunable by the Adaptive Engine on page 195
- Configuring a PMA Parameter Using Native PHY IP on page 198

9.2.24.1. Reading PMA Analog Parameters

Use attribute code 0x002C to read the analog parameter. Refer to the following table for information on how to read the value.

Parameter Name	{0x85[7:0],0x84[7:0]}
GainLF	0x0901
CTLE LF Min	0x0909
CTLE LF Max	0x090A
GainHF	0x0900
CTLE HF min	0x0907
CTLE HF max	0x0908
GS1	0x0904
GS2	0x0905
RF_P2	0x0D00
RF_P2_MIN	0x0D09
RF_P2_MAX	0x0D0A
RF_P1	0x0D01
RF_P1_MIN	0x0D07
RF_P1_MAX	0x0D08
RF_P0	0x0D02
RF_B1	0x0D04
RF_B0	0x0D03
RF_B0T	0x200B
RF_A	{0x85[7:0],0x84[7:0]} 0x2001
	{0x85[7:0],0x84[7:0]} 0x2002

If you need to change the RF_A parameter, update both sets of RF_A parameters:

1. Use attribute code 0x002C with {0x85[7:0],0x84[7:0]} = 0x2001 to read the first parameter.
2. Use attribute code 0x006C to update the first parameter.
3. Use attribute code 0x002C with {0x85[7:0],0x84[7:0]} = 0x2002 to read the second parameter.
4. Use attribute code 0x006C to update the second parameter.

Note: The above steps are a high level description of the steps. Please refer to *PMA Attribute Details* for details on how to send the PMA attributes on the AVMM bus.

The value is returned on AVMM registers {0x89[7:0],0x88[7:0]}.

Related Information

PMA Attribute Details on page 161

9.2.24.2. Updating PMA Analog Parameters

To write a new value to an analog parameter for the RX adaptation to use as the starting value:

1. Use attribute code 0x002C to read the parameter value.
2. Use attribute code 0x006C to enter the new value on {0x85[7:0],0x84[7:0]} as a two's complement number.

The analog parameter written is determined by the last previous analog parameter read. Registers {0x89[7:0],0x88[7:0]} return 0x002C to indicate that the update was successful.

9.2.24.3. Loading Parameters into the Receiver

After updating the analog parameter value, the parameters listed below need to be loaded into the receiver to become effective. Use attribute code 0x00EC and the following table to load the parameters into the transceiver.

Parameters to load	{0x85[7:0],0x84[7:0]}
GainLF, GainHF, GS1 and GS2	0x0015
RF_P2, RF_P1, RF_P0, RF_B1, RF_B0	0x0012

9.2.24.4. Fixing Parameter Values

For optimal transceiver performance in certain data rate and channel profiles, some analog parameters should be fixed so that subsequent initial coarse adaptation or continuous adaptation does not change the value.

To fix the GainLF parameter, use attribute code 0x002C and 0x006C in sequence.

1. Use attribute code 0x002C and {0x85[7:0],0x84[7:0]} set to 0x0108.

   The return value on registers {0x89[7:0],0x88[7:0]} is the current fix status.
2. Use attribute code 0x006C with the following values:

- 0x84[0] = 1'b0
- 0x84[1] = 1'b1 to fix GainLF or 1'b0 to allow adaptation to set GainLF
- 0x84[3:2] = 2'b00
- 0x84[4] = 1'b1 to fix GS2 or 1'b0 to allow adaptation to set GS2
- 0x84[5] = 1'b1 to fix GS1 or 1'b0 to allow adaptation to set GS1
- 0x84[7:6] = 2'b00
- 0x85[7:0] = 8'h00

The return value on registers {0x89[7:0],0x88[7:0]} is 0x002C.

To fix the RF_P2, RF_B1, and RF_B0 parameters, use attribute code 0x002C and 0x006C in sequence.

1. Use attribute code 0x002C and {0x85[7:0],0x84[7:0]} set to 0x0109.

   The return value on registers {0x89[7:0],0x88[7:0]} is the current fix status.
2. Use attribute code 0x006C with the following values:
   - 0x84[0] = 1'b1 to fix RF_P2 or 1'b0 to allow adaptation to set RF_P2
   - 0x84[2:1] = 2'b00
   - 0x84[3] = 1'b1 to fix RF_B0 or 1'b0 to allow adaptation to set RF_B0.
   - 0x84[4] = 1'b1 to fix RF_B1 or 1'b0 to allow adaptation to set RF_B1.
   - 0x84[7:5] = 3'h0
   - 0x85[7:0] = 8'h00

   The return value on registers {0x89[7:0],0x88[7:0]} is 0x002C.

9.2.24.5. Reading NRZ/PAM4 Eye Height

Use attribute code 0x002C to read the NRZ/PAM4 eye height.

For NRZ, all six of the eye heights below return the same value, so you can use any one of them to get the eye height.

For PAM4, the six eye heights return different values because PAM4 uses all six.

The eye height is returned on registers {0x89[7:0],0x88[7:0]}.

Table 75. Reading NRZ/PAM4 Eye Height

Parameter Name	Value
Lower Even Eye	{0x85[7:0],0x84[7:0]}: 0x1700
Middle Even Eye	{0x85[7:0],0x84[7:0]: 0x1701
Upper Even Eye	{0x85[7:0],0x84[7:0]: 0x1702
Lower Odd Eye	{0x85[7:0],0x84[7:0]}: 0x1703
Middle Odd Eye	{0x85[7:0],0x84[7:0]: 0x1704
Upper Odd Eye	{0x85[7:0],0x84[7:0]: 0x1705

9.2.24.6. Enabling and Disabling Electrical Idle Detector Filtering and Reading Electrical Idle Detector Status

Electrical Idle Detector Filtering Controls

The `signal_ok_config` 16-bit register controls electrical idle detector filtering.

To read:

1. Use attribute code 0x002C and {0x85[7:0],0x84[7:0]}: 0x11C. The value is returned on registers {0x89[7:0],0x88[7:0]}.

To write:

1. Use attribute code 0x002C and {0x85[7:0],0x84[7:0]}: 0x11C.
2. Use attribute code 0x006C and {0x85[7:0],0x84[7:0]}:
   - 0x84[0] to enable 0x4027[4] to track electrical idle instead of its normal use.
   - 0x84[1] to cause 0x4027[4] to oscillate when a signal is detected.
   - 0x84[2] to enable DFE tuning based electrical idle detector detection
   - 0x84[3] is reserved
   - 0x84[7:4], 0x85[7:0], if 0x84[2] is 0, to set the number of consecutive polling loops of electrical idle that it must be high before 0x4027[4] goes high. If 0x84[0] is 0, there is no effect.
   - 0x84[7:4], 0x85[7:0], if 0x84[2] is 1, to set the eye height threshold for DFE based on the electrical idle detector. If 0x84[0] is 0, there is no effect.

Enabling and Disabling Electrical Idle Detector Filtering

To enable electrical idle detector filtering:

1. Use attribute code 0x002C and {0x85[7:0],0x84[7:0]}: 0x11C.
2. Use attribute code 0x006C and {0x85[7:0],0x84[7:0]}: 0xFFF1.

To enable electrical idle detector filtering based on DFE tuning:

1. Use attribute code 0x002C and {0x85[7:0],0x84[7:0]}: 0x11C.
2. Use attribute code 0x006C and {0x85[7:0],0x84[7:0]}: 0x0155.

To disable any mode of electrical idle detector filtering:

1. Use attribute code 0x002C and {0x85[7:0],0x84[7:0]}: 0x11C.
2. Use attribute code 0x006C and {0x85[7:0],0x84[7:0]}: 0x0.

Reading Electrical Idle Detector Status

After electrical idle detector filtering is enabled, to read the electrical idle detector status:

- Use attribute code 0x4027 and {0x85[7:0],0x84[7:0]}: 0x0. The value is returned on registers {0x89[7:0],0x88[7:0]}: 0x30 for no signal and 0x20 for signal present.

9.2.24.7. Initial Adaptation Effort Levels

To read:

1. Use attribute code 0x002C and {0x85[7:0],0x84[7:0]}: 0x0118. The value is returned on registers {0x89[7:0],0x88[7:0]}.

To write:

1. Use attribute code 0x002C and {0x85[7:0],0x84[7:0]}: 0x0118.
2. Use attribute code 0x006C and {0x85[7:0],0x84[7:0]}:
   - 0x84[1:0] = 0x00 for low effort (`00_effort`), 0x10 for medium effort (`05_effort`), 0x01 for full effort (`10_effort`)
   - 0x84[7:2], 0x85[7:0] is reserved

Low effort initial adaptation can be used for NRZ only. Medium effort initial adaptation can be used for PAM4 only. Full effort initial adaptation can be used for NRZ or PAM4.

Low effort initial adaptation is the quickest to complete and is recommended for NRZ Ethernet AN/LT and CPRI protocols.

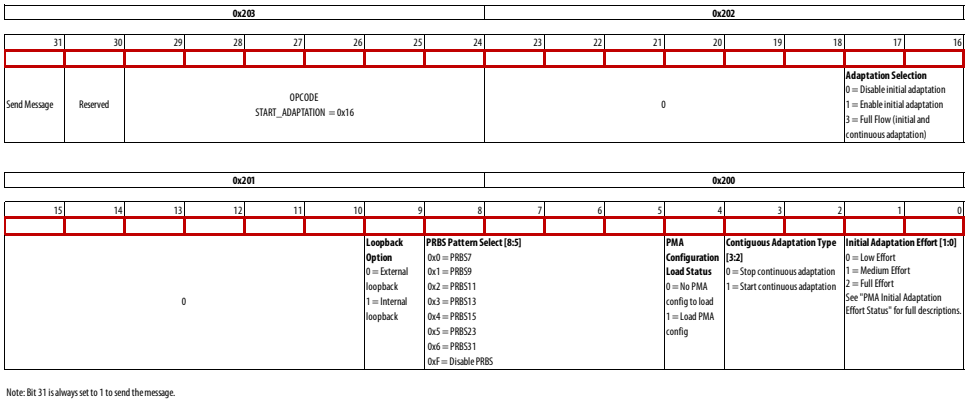
Medium effort initial adaptation is recommended for PAM4 Ethernet AN/LT.

Full effort initial adaptation is for general usage and provides the best performance and stability. This is the recommended, and default, adaptation mode.

9.3. PMA Registers 0x200 to 0x203 Usage

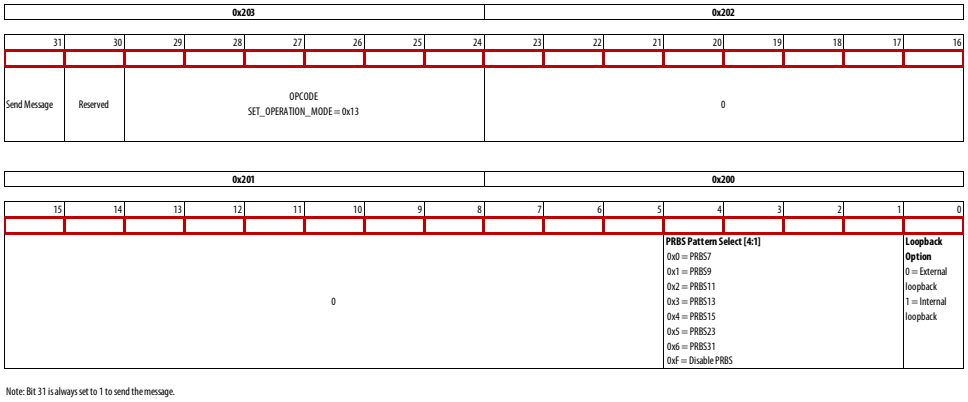
Use registers 0x200 to 0x203 as an alternative method to set certain PMA attributes or to perform a PMA analog reset. For details, refer to *Loading a PMA Configuration*.

Figure 116. Loading PMA Configuration Register START_ADAPTATION



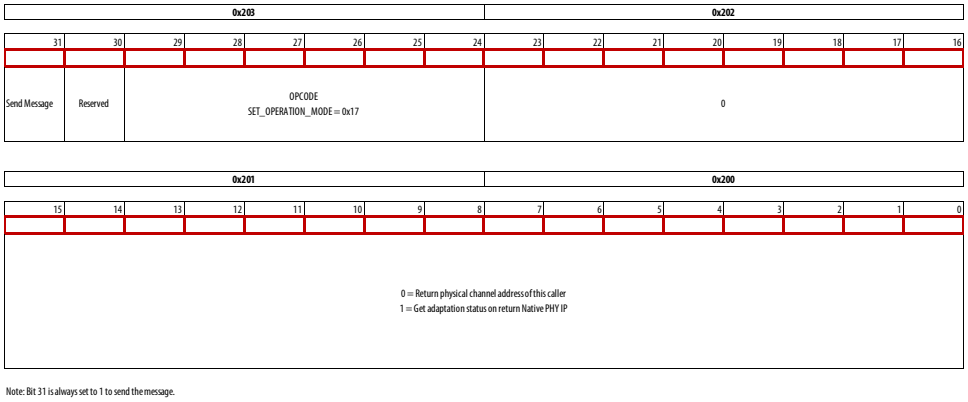
Note: Bit 31 is always set to 1 to send the message.

Figure 117. Loading PMA Configuration Register SET_OPERATION_MODE



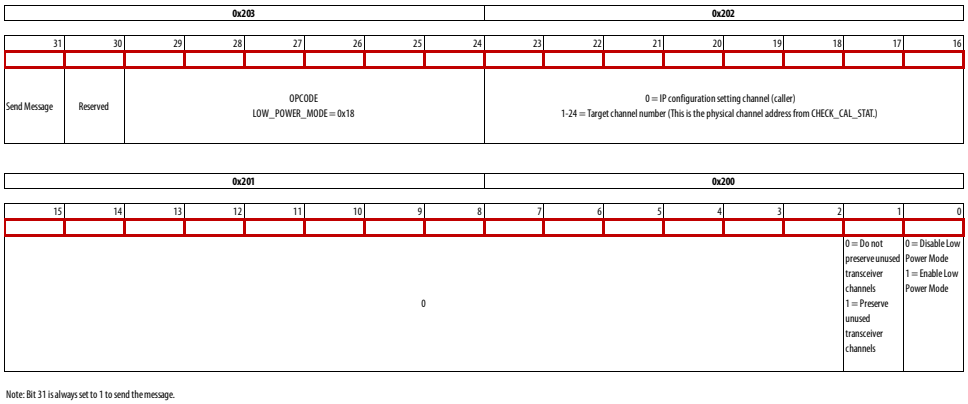
Note: Bit 31 is always set to 1 to send the message.

Figure 118. Loading PMA Configuration Register CHECK_CAL_STAT



Note: Bit 31 is always set to 1 to send the message.

Figure 119. Loading PMA Configuration Register LOW_POWER_MODE

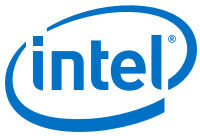


Note: Bit 31 is always set to 1 to send the message.

Refer to *PMA Analog Reset*, *Set PRBS Mode and Internal Serial Loopback*, *Start Adaptation and Put PMA in Mission Mode*, *Read the Physical Channel Number*, and *Check the PMA Adaptation Status* for how to set registers 0x200 to 0x203.

Related Information

- Loading a PMA Configuration on page 204



- PMA Analog Reset on page 237
- Set PRBS Mode and Internal Serial Loopback on page 237
- Start Adaptation and Put PMA in Mission Mode on page 237
- Read the Physical Channel Number on page 238
- Check the PMA Adaptation Status on page 238
- RX Adaptation Modes on page 78
  See "PMA Initial Adaptation Effort Status."

9.3.1. PMA Analog Reset

Register 0x200: 0x00

Register 0x201: 0x00

Register 0x202: 0x00

Register 0x203: 0x81

Related Information

PMA Analog Reset on page 132

9.3.2. Set PRBS Mode and Internal Serial Loopback

This configures both the PRBS generator and checker for different PRBS patterns.

- 0x200[0]: 1'b1 to enable internal serial loopback enabled. 1'b0 to disable internal serial loopback.
- 0x200[4:1]: 4'h0 for PRBS7
- 0x200[4:1]: 4'h1 for PRBS9
- 0x200[4:1]: 4'h2 for PRBS11
- 0x200[4:1]: 4'h3 for PRBS13
- 0x200[4:1]: 4'h4 for PRBS15
- 0x200[4:1]: 4'h5 for PRBS23
- 0x200[4:1]: 4'h6 for PRBS31
- 0x200[4:1] 4'hF to disable PRBS
- 0x201[7:0]: 8'h00
- 0x202[7:0]: 8'h00
- 0x203[7:0]: 8'h93

9.3.3. Start Adaptation and Put PMA in Mission Mode

You can start initial and continuous adaptation and place the transceiver in mission mode.

- 0x200[4:0]: 4'h6
- {0x201[0],0x200[7:5]}: 4'h0 to set the PRBS generator and checker in PRBS7 mode after initial adaption is complete.
- {0x201[0],0x200[7:5]}: 4'h1 to set the PRBS generator and checker in PRBS9 mode after initial adaption is complete.
- {0x201[0],0x200[7:5]}: 4'h2 to set the PRBS generator and checker in PRBS11 mode after initial adaption is complete.
- {0x201[0],0x200[7:5]}: 4'h3 to set the PRBS generator and checker in PRBS13 mode after initial adaption is complete.
- {0x201[0],0x200[7:5]}: 4'h4 to set the PRBS generator and checker in PRBS15 mode after initial adaption is complete.
- {0x201[0],0x200[7:5]}: 4'h5 to set the PRBS generator and checker in PRBS23 mode after initial adaption is complete.
- {0x201[0],0x200[7:5]}: 4'h6 to set the PRBS generator and checker in PRBS31 mode after initial adaption is complete.
- {0x201[0],0x200[7:5]}: 4'hF to disable the PRBS generator and checker after initial adaptation is complete. This places the PMA in mission mode.
- 0x201[1]: 1'b1 to enable internal serial loopback after initial adaptation is completed.
- 0x201[1]: 1'b0 to disable internal serial loopback after initial adaptation is completed.
- 0x201[7:2]: 6'h00
- 0x202[7:0]: 8'h03
- 0x203[7:0]: 0x96

9.3.4. Read the Physical Channel Number

To read the physical channel number of the current channel on the E-Tile:

- 0x200[7:0]: 8'h00
- 0x201[7:0]: 8'h00
- 0x202[7:0]: 8'h00
- 0x203[7:0]: 8'h97

After writing to register 0x203, read 0x207 until it becomes 0x80. This indicates that the operation completed successfully. The physical channel number is located on register 0x204.

9.3.5. Check the PMA Adaptation Status

To read the PMA adaptation status of the current channel on the E-tile:

- 0x200[7:0]: 8'h01
- 0x201[7:0]: 8'h00
- 0x202[7:0]: 8'h00
- 0x203[7:0]: 8'h97

After writing to register 0x203, read 0x207 until it becomes 0x80. This indicates that the operation completed successfully. The status is located on register 0x204 and can be interpreted as follows:

- Initial adaptation done: 0x80
- Initial adaptation in progress: 0x91
- One time continuous adaptation done: 0x80
- One time continuous adaptation in progress: 0xA2
- Running continuous adaptation: 0xE2

9.3.6. Load a PMA Configuration

This command selects internal serial loopback, PRBS31, and loads a PMA configuration. It does not perform any PMA adaptations.

1. Load a PMA configuration to the current channel on the E-tile:
   a. Write 0x200[7:0] = 8'hD0.
   b. Write 0x201[7:0] = 8'h02.
   c. Write 0x202[7:0] = 8'h00.
   d. Write 0x203[7:0] = 8'h96.
2. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.

9.4. Supported Data Rate Ratios for PMA Attribute Codes 0x0005 and 0x0006

Use the following table to set registers 0x85[7:0] down to 0x84[7:0] when issuing either PMA attribute code 0x0005 or 0x0006.

Table 76. Supported Baud Rate Ratios

Ratio	Supported below 15G	Supported above 15G	1/2 Rate	1/4 Rate	1/8 Rate
5	Yes	No	No	No	No
8	Yes	No	No	No	No
9	Yes	No	No	No	No
10	Yes	No	Yes	No	No
11	Yes	No	No	No	No
12	Yes	No	No	No	No
13	Yes	No	No	No	No
14	Yes	No	No	No	No
15	Yes	No	No	No	No
16	Yes	No	Yes	No	No
17	Yes	No	No	No	No
18	Yes	No	Yes	NO	No

continued...

Ratio	Supported below 15G	Supported above 15G	1/2 Rate	1/4 Rate	1/8 Rate
19	Yes	No	No	No	No
20	Yes	No	Yes	Yes	No
21	Yes	No	No	No	No
22	Yes	No	Yes	No	No
23	Yes	No	No	No	No
24	Yes	No	Yes	No	No
25	Yes	No	No	No	No
26	Yes	No	Yes	No	No
27	Yes	No	No	No	No
28	Yes	Yes	Yes	No	No
29	Yes	No	No	No	No
30	Yes	Yes	Yes	No	No
31	Yes	No	No	No	No
32	Yes	Yes	Yes	Yes	No
33	Yes	No	No	No	No
34	Yes	Yes	Yes	No	No
35	Yes	Yes	No	No	No
36	Yes	Yes	Yes	Yes	No
37	Yes	No	No	No	No
38	Yes	Yes	Yes	No	No
39	Yes	No	No	No	No
40	Yes	Yes	Yes	Yes	Yes
41	Yes	No	No	No	No
42	Yes	Yes	Yes	No	No
43	Yes	No	No	No	No
44	Yes	Yes	Yes	Yes	No
45	Yes	Yes	No	No	No
46	Yes	Yes	Yes	No	No
47	Yes	No	No	No	No
48	Yes	Yes	Yes	Yes	No
49	Yes	No	No	No	No
50	Yes	Yes	Yes	No	No
51	Yes	No	No	No	No
52	Yes	Yes	Yes	Yes	No
53	Yes	No	No	No	No
54	Yes	Yes	Yes	No	No

continued...

Ratio	Supported below 15G	Supported above 15G	1/2 Rate	1/4 Rate	1/8 Rate
55	Yes	Yes	No	No	No
56	Yes	Yes	Yes	Yes	No
57	Yes	No	No	No	No
58	Yes	Yes	Yes	No	No
59	Yes	No	No	No	No
60	Yes	Yes	Yes	Yes	No
61	Yes	No	No	No	No
62	Yes	Yes	Yes	No	No
63	Yes	No	No	No	No
64	Yes	Yes	Yes	Yes	Yes
65	Yes	Yes	No	No	No
66	Yes	Yes	Yes	No	No
67	Yes	No	No	No	No
68	Yes	Yes	Yes	Yes	No
69	Yes	No	No	No	No
70	Yes	Yes	Yes	No	No
71	Yes	No	No	No	No
72	Yes	Yes	Yes	Yes	Yes
73	Yes	No	No	No	No
74	Yes	Yes	Yes	No	No
75	Yes	Yes	No	No	No
76	Yes	Yes	Yes	Yes	No
77	Yes	No	No	No	No
78	Yes	Yes	Yes	No	No
79	Yes	No	No	No	No
80	Yes	Yes	Yes	Yes	Yes
81	Yes	No	No	No	No
82	Yes	Yes	Yes	No	No
83	Yes	No	No	No	No
84	Yes	Yes	Yes	Yes	No
85	Yes	Yes	No	No	No
86	Yes	Yes	Yes	No	No
87	Yes	No	No	No	No
88	Yes	Yes	Yes	Yes	Yes
89	Yes	No	No	No	No
90	Yes	Yes	Yes	No	No

continued...

Ratio	Supported below 15G	Supported above 15G	1/2 Rate	1/4 Rate	1/8 Rate
91	Yes	No	No	No	No
92	Yes	Yes	Yes	Yes	No
93	Yes	No	No	No	No
94	Yes	Yes	Yes	No	No
95	Yes	Yes	No	No	No
96	Yes	Yes	Yes	Yes	Yes
97	Yes	No	No	No	No
98	Yes	Yes	Yes	No	No
99	Yes	No	No	No	No
100	Yes	Yes	Yes	Yes	No
101	Yes	No	No	No	No
102	Yes	Yes	Yes	No	No
103	Yes	No	No	No	No
104	Yes	Yes	Yes	Yes	Yes
105	Yes	Yes	No	No	No
106	Yes	Yes	Yes	No	No
107	Yes	No	No	No	No
108	Yes	Yes	Yes	Yes	No
109	Yes	No	No	No	No
110	Yes	Yes	Yes	No	No
111	Yes	No	No	No	No
112	Yes	Yes	Yes	Yes	Yes
113	Yes	No	No	No	No
114	Yes	Yes	Yes	No	No
115	Yes	Yes	No	No	No
116	Yes	Yes	Yes	Yes	No
117	Yes	No	No	No	No
118	Yes	Yes	Yes	No	No
119	Yes	No	No	No	No
120	Yes	Yes	Yes	Yes	Yes
121	Yes	No	No	No	No
122	Yes	Yes	Yes	No	No
123	Yes	No	No	No	No
124	Yes	Yes	Yes	Yes	No
125	Yes	Yes	No	No	No
126	Yes	Yes	Yes	No	No

continued...

Ratio	Supported below 15G	Supported above 15G	1/2 Rate	1/4 Rate	1/8 Rate
127	Yes	No	No	No	No
128	Yes	Yes	Yes	Yes	Yes
129	Yes	No	No	No	No
130	Yes	Yes	Yes	No	No
131	Yes	No	No	No	No
132	Yes	Yes	Yes	Yes	No
133	Yes	No	No	No	No
134	Yes	Yes	Yes	No	No
135	Yes	Yes	No	No	No
136	Yes	Yes	Yes	Yes	Yes
137	Yes	No	No	No	No
138	Yes	Yes	Yes	No	No
139	Yes	No	No	No	No
140	Yes	Yes	Yes	Yes	No
141	Yes	No	No	No	No
142	Yes	Yes	Yes	No	No
143	Yes	No	No	No	No
144	Yes	Yes	Yes	Yes	Yes
145	Yes	Yes	No	No	No
146	Yes	Yes	Yes	No	No
147	Yes	No	No	No	No
148	Yes	Yes	Yes	Yes	No
149	Yes	No	No	No	No
150	Yes	Yes	Yes	No	No
151	Yes	No	No	No	No
152	Yes	Yes	Yes	Yes	Yes
153	Yes	No	No	No	No
154	Yes	Yes	Yes	No	No
155	Yes	Yes	No	No	No
156	Yes	Yes	Yes	Yes	No
157	Yes	No	No	No	No
158	Yes	Yes	Yes	No	No
159	Yes	No	No	No	No
160	Yes	Yes	Yes	Yes	Yes
161	Yes	No	No	No	No
162	Yes	Yes	Yes	No	No

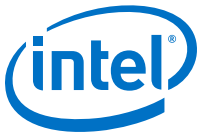
continued...

Ratio	Supported below 15G	Supported above 15G	1/2 Rate	1/4 Rate	1/8 Rate
163	Yes	No	No	No	No
164	Yes	Yes	Yes	Yes	No
165	Yes	Yes	No	No	No
166	Yes	Yes	Yes	No	No
167	Yes	No	No	No	No
168	Yes	Yes	Yes	Yes	Yes
169	Yes	No	No	No	No
170	Yes	Yes	Yes	No	No
171	Yes	No	No	No	No
172	Yes	Yes	Yes	Yes	No
173	Yes	No	No	No	No
174	Yes	Yes	Yes	No	No
175	Yes	Yes	No	No	No
176	Yes	Yes	Yes	Yes	Yes
177	Yes	No	No	No	No
178	Yes	Yes	Yes	No	No
179	Yes	No	No	No	No
180	Yes	Yes	Yes	Yes	No

Table 77. Data Rate Ratios for PMA Attribute 0x0005 or 0x0006

Ratio	Supported Above 15 Gbaud per Second	1/2 Rate	1/4 Rate	1/8 Rate
10	No	Yes	No	No
16	No	Yes	No	No
18	No	Yes	No	No
20	No	Yes	Yes	No
22	No	Yes	No	No
24	No	Yes	No	No
26	No	Yes	No	No
28	Yes	Yes	No	No
30	Yes	Yes	No	No
32	Yes	Yes	Yes	No
34	Yes	Yes	No	No
35	Yes	No	No	No
36	Yes	Yes	Yes	No
38	Yes	Yes	No	No
40	Yes	Yes	Yes	Yes

continued...



Ratio	Supported Above 15 Gbaud per Second	1/2 Rate	1/4 Rate	1/8 Rate
42	Yes	Yes	No	No
44	Yes	Yes	Yes	No
46	Yes	Yes	No	No
48	Yes	Yes	Yes	No
37	No	No	No	No
50	Yes	Yes	No	No
39	No	No	No	No
52	Yes	Yes	Yes	No
54	Yes	Yes	No	No
55	Yes	No	No	No
56	Yes	Yes	Yes	No
58	Yes	Yes	No	No
60	Yes	Yes	Yes	No
62	Yes	Yes	No	No
64	Yes	Yes	Yes	Yes
65	Yes	No	No	No
66	Yes	Yes	No	No
68	Yes	Yes	Yes	No
70	Yes	Yes	No	No
72	Yes	Yes	Yes	Yes
74	Yes	Yes	No	No
75	Yes	No	No	No
76	Yes	Yes	Yes	No
78	Yes	Yes	No	No
80	Yes	Yes	Yes	Yes
82	Yes	Yes	No	No
84	Yes	Yes	Yes	No
85	Yes	No	No	No
86	Yes	Yes	No	No
88	Yes	Yes	Yes	Yes
90	Yes	Yes	No	No
92	Yes	Yes	Yes	No
94	Yes	Yes	No	No
95	Yes	No	No	No
96	Yes	Yes	Yes	Yes

continued...

Ratio	Supported Above 15 Gbaud per Second	1/2 Rate	1/4 Rate	1/8 Rate
98	Yes	Yes	No	No
100	Yes	Yes	Yes	No
102	Yes	Yes	No	No
104	Yes	Yes	Yes	Yes
106	Yes	Yes	No	No
108	Yes	Yes	Yes	No
110	Yes	Yes	No	No
112	Yes	Yes	Yes	Yes
114	Yes	Yes	No	No
116	Yes	Yes	Yes	No
118	Yes	Yes	No	No
120	Yes	Yes	Yes	Yes
122	Yes	Yes	No	No
124	Yes	Yes	Yes	No
125	Yes	No	No	No
126	Yes	Yes	No	No
128	Yes	Yes	Yes	Yes
130	Yes	Yes	No	No
132	Yes	Yes	Yes	No
134	Yes	Yes	No	No
135	Yes	No	No	No
136	Yes	Yes	Yes	Yes
138	Yes	Yes	No	No
139	No	No	No	No
140	Yes	Yes	Yes	No
142	Yes	Yes	No	No
144	Yes	Yes	Yes	Yes
145	Yes	No	No	No
146	Yes	Yes	No	No
148	Yes	Yes	Yes	No
150	Yes	Yes	No	No
152	Yes	Yes	Yes	Yes
154	Yes	Yes	No	No
155	Yes	No	No	No
156	Yes	Yes	Yes	No

continued...

Ratio	Supported Above 15 Gbaud per Second	1/2 Rate	1/4 Rate	1/8 Rate
158	Yes	Yes	No	No
160	Yes	Yes	Yes	Yes
162	Yes	Yes	No	No
164	Yes	Yes	Yes	No
165	Yes	No	No	No
166	Yes	Yes	No	No
168	Yes	Yes	Yes	Yes
170	Yes	Yes	No	No
172	Yes	Yes	Yes	No
174	Yes	Yes	No	No
175	Yes	No	No	No
176	Yes	Yes	Yes	Yes
178	Yes	Yes	No	No
180	Yes	Yes	Yes	No

9.5. RS-FEC Registers

The delay between RS-FEC register reads should be at least 10 µs. There is no such requirement for register writes. As a result, RS-FEC register back-to-back writes are allowed.

In order to get reliable readouts of the registers with addresses greater than 0x100, ensure RS-FEC has a valid clock.

Reconfiguration and status monitoring of the RS-FEC block in the E-tile Native PHY IP core is provided through a dedicated Avalon Memory-Mapped (AVMM) interface. This interface specifically reads and writes the control and status registers associated with the RS-FEC block. A separate AVMM interface, called the channel AVMM interface, reads and writes the control and status registers associated with the other blocks in the E-tile Native PHY IP core.

The AVMM interface for the RS-FEC consists of eight interface ports exposed at the top level of the Native PHY IP core.

Table 78. RS-FEC AVMM Interface Ports

input	wire		reconfig_rsfec_clk
input	wire		reconfig_rsfec_reset
input	wire		reconfig_rsfec_write
input	wire		reconfig_rsfec_read
input	wire	[10:0]	reconfig_rsfec_address

continued...

input	wire	[7:0]	reconfig_rsfec_writedata
output	wire	[7:0]	reconfig_rsfec_readdata
output	wire		reconfig_rsfec_waitrequest

The reconfig_rsfec_clk input must be driven by a 100-125 MHz clock. The same clock can be used for this input as for the channel AVMM interface.

The reconfig_rsfec_reset input must be asserted for at least one clock cycle of the reconfiguration clock after the device has entered user mode. The Reset IP can be used to provide this reset signal.

The reconfig_rsfec_write and reconfig_rsfec_read inputs are used to indicate to the AVMM interface whether a read or write operation is desired. The reconfig_rsfec_address input is used for both read and write operations and indicates which register address is targeted during the current AVMM access operation.

The reconfig_rsfec_writedata and reconfig_rsfec_readdata contain the data to be written to or the data read from the register address set by the 11 bit reconfig_rsfec_address input.

The reconfig_rsfec_writedata and reconfig_rsfec_readdata ports are 8 bits wide. The RS-FEC registers, in the hardware, are 32 bits wide, but they use 4 byte increment addressing, so the 32 bit register can be addressed as four 8 bit registers with contiguous byte addresses.

For example, the least-significant byte (lowest order byte) of register 0x04 (where register address 0x04 is considered as a 32 bit register address) is accessed by reading 8 bits from or writing 8 bits to address 0x04. To get the next most significant byte of the same 32 bit register, read 8 bits from or write 8 bits to address 0x05. To get the rest of the register, use addresses 0x06 and 0x07. The next 32 bit register has an address of 0x08. In other words, the 32 bit register addresses increment by 4 bytes for each register.

Effectively, the registers are addressed, read, and written one byte at a time. This matches the hardware implementation of the AVMM interface.

The reconfig_rsfec_waitrequest output asserts when the AVMM interface is busy servicing an AVMM operation and deasserts when the AVMM is available for the next operation.

Table 79. RS-FEC Registers

Address	Name	Description	Reset
0x04	rsfec_top_clk_cfg	RS-FEC Clock configuration register	0x0000 0F00
0x10	rsfec_top_tx_cfg	RS-FEC TX configuration register	0x0000 0000
0x14	rsfec_top_rx_cfg	RS-FEC RX configuration register	0x0000 0000
0x20	tx_aib_dsk_conf	Defines the configuration fields for TX Deskew	0x0000 0000
0x30	rsfec_core_cfg	RS-FEC core configuration	0x0000 0000
0x40	rsfec_lane_cfg_0	RS-FEC per lane configuration	0x0000 0000
0x44	rsfec_lane_cfg_1		

continued...

Address	Name	Description	Reset
0x48	rsfec_lane_cfg_2		
0x4C	rsfec_lane_cfg_3		
0x104	tx_aib_dsk_status	Status fields for TX Deskew	0x0000 0000
0x108	rsfec_debug_cfg	Extra config/debug on fec_clock	0x0000 0000
0x120	rsfec_lane_tx_stat_0	RS-FEC per lane TX status	0x0000 0000
0x124	rsfec_lane_tx_stat_1		
0x128	rsfec_lane_tx_stat_2		
0x12C	rsfec_lane_tx_stat_3		
0x130	rsfec_lane_tx_hold_0	RS-FEC per lane TX status hold	0x0000 0000
0x134	rsfec_lane_tx_hold_1		
0x138	rsfec_lane_tx_hold_2		
0x13C	rsfec_lane_tx_hold_3		
0x140	rsfec_lane_tx_inten_0	RS-FEC per lane TX status hold interrupt - set to 1 to enable rsfec_lane_tx lane interrupt	0x0000 0000
0x144	rsfec_lane_tx_inten_1		
0x148	rsfec_lane_tx_inten_2		
0x14C	rsfec_lane_tx_inten_3		
0x150	rsfec_lane_rx_stat_0	RS-FEC per lane RX status	0x0000 0000
0x154	rsfec_lane_rx_stat_1		
0x158	rsfec_lane_rx_stat_2		
0x15C	rsfec_lane_rx_stat_3		
0x160	rsfec_lane_rx_hold_0	RS-FEC per lane RX status hold	0x0000 0000
0x164	rsfec_lane_rx_hold_1		
0x168	rsfec_lane_rx_hold_2		
0x16C	rsfec_lane_rx_hold_3		
0x170	rsfec_lane_rx_inten_0	RS-FEC per lane RX status hold interrupt - set to 1 to enable rsfec_lane_rx lane interrupt	0x0000 0000
0x174	rsfec_lane_rx_inten_1		
0x178	rsfec_lane_rx_inten_2		
0x17C	rsfec_lane_rx_inten_3		
0x180	rsfec_lanes_rx_stat	RS-FEC combined lanes RX status	0x0000 0000
0x188	rsfec_lanes_rx_hold	RS-FEC combined lanes RX hold status	0x0000 0000
0x18C	rsfec_lanes_rx_inten	RS-FEC combined lanes RX interrupt enable - set to 1 to enable rsfec_lanes RX lane interrupt	0x0000 0000
0x1A0	rsfec_ln_mapping_rx_0	RS-FEC FEC lane mapping	0x0000 0000
0x1A4	rsfec_ln_mapping_rx_1		
0x1A8	rsfec_ln_mapping_rx_2		
0x1AC	rsfec_ln_mapping_rx_3		

continued...

Address	Name	Description	Reset
0x1B0	rsfec_ln_skew_rx_0	RS-FEC lane skew	0x0000 0000
0x1B4	rsfec_ln_skew_rx_1		
0x1B8	rsfec_ln_skew_rx_2		
0x1BC	rsfec_ln_skew_rx_3		
0x1C0	rsfec_cw_pos_rx_0	RS-FEC codeword bit position on RX	0x0000 0000
0x1C4	rsfec_cw_pos_rx_1		
0x1C8	rsfec_cw_pos_rx_2		
0x1CC	rsfec_cw_pos_rx_3		
0x1D0	rsfec_core_ecc_hold	RS-FEC SRAM ECC status hold	0x0000 0000
0x1E0	rsfec_err_inj_tx_0	RS-FEC error injection mode	0x0000 0000
0x1E4	rsfec_err_inj_tx_1		
0x1E8	rsfec_err_inj_tx_2		
0x1EC	rsfec_err_inj_tx_3		
0x1F0	rsfec_err_val_tx_0	RS-FEC per lane error injection status	0x0000 0000
0x1F4	rsfec_err_val_tx_1		
0x1F8	rsfec_err_val_tx_2		
0x1FC	rsfec_err_val_tx_3		
0x200	rsfec_corr_cw_cnt_0_lo	RS-FEC number of FEC codewords with errors that were corrected (low word: bits 31 to 0)	0x0000 0000
0x208	rsfec_corr_cw_cnt_1_lo		
0x210	rsfec_corr_cw_cnt_2_lo		
0x218	rsfec_corr_cw_cnt_3_lo		
0x204	rsfec_corr_cw_cnt_0_hi	RS-FEC number of FEC codewords with errors that were corrected (high word: bits 63 to 32)	0x0000 0000
0x20C	rsfec_corr_cw_cnt_1_hi		
0x214	rsfec_corr_cw_cnt_2_hi		
0x21C	rsfec_corr_cw_cnt_3_hi		
0x220	rsfec_uncorr_cw_cnt_0_lo	RS-FEC number of FEC codewords that could not be corrected due to too many errors (low word: bits 31 to 0)	0x0000 0000
0x228	rsfec_uncorr_cw_cnt_1_lo		
0x230	rsfec_uncorr_cw_cnt_2_lo		
0x238	rsfec_uncorr_cw_cnt_3_lo		
0x224	rsfec_uncorr_cw_cnt_0_hi	RS-FEC number of FEC codewords that could not be corrected due to too many errors (high word: bits 63 to 32)	0x0000 0000
0x22C	rsfec_uncorr_cw_cnt_1_hi		
0x234	rsfec_uncorr_cw_cnt_2_hi		
0x23C	rsfec_uncorr_cw_cnt_3_hi		
0x240	rsfec_corr_syms_cnt_0_lo	RS-FEC number of 10b symbols corrected for the lane (low word: bits 31 to 0)	0x0000 0000
0x248	rsfec_corr_syms_cnt_1_lo		
0x250	rsfec_corr_syms_cnt_2_lo		

continued...

Address	Name	Description	Reset
0x258	rsfec_corr_syms_cnt_3_lo		
0x244	rsfec_corr_syms_cnt_0_hi	RS-FEC number of 10b symbols corrected for the lane (high word: bits 63 to 32)	0x0000 0000
0x24C	rsfec_corr_syms_cnt_1_hi		
0x254	rsfec_corr_syms_cnt_2_hi		
0x25C	rsfec_corr_syms_cnt_3_hi		
0x260	rsfec_corr_0s_cnt_0_lo	RS-FEC number of bits corrected 0->1 for the lane (low word: bits 31 to 0)	0x0000 0000
0x268	rsfec_corr_0s_cnt_1_lo		
0x270	rsfec_corr_0s_cnt_2_lo		
0x278	rsfec_corr_0s_cnt_3_lo		
0x264	rsfec_corr_0s_cnt_0_hi	RS-FEC number of bits corrected 0->1 for the lane (high word: bits 63 to 32)	0x0000 0000
0x26C	rsfec_corr_0s_cnt_1_hi		
0x274	rsfec_corr_0s_cnt_2_hi		
0x27C	rsfec_corr_0s_cnt_3_hi		
0x280	rsfec_corr_1s_cnt_0_lo	RS-FEC number of bits corrected 1->0 for the lane (low word: bits 31 to 0)	0x0000 0000
0x288	rsfec_corr_1s_cnt_1_lo		
0x290	rsfec_corr_1s_cnt_2_lo		
0x298	rsfec_corr_1s_cnt_3_lo		
0x284	rsfec_corr_1s_cnt_0_hi	RS-FEC number of bits corrected 1->0 for the lane (high word: bits 63 to 32)	0x0000 0000
0x28C	rsfec_corr_1s_cnt_1_hi		
0x294	rsfec_corr_1s_cnt_2_hi		
0x29C	rsfec_corr_1s_cnt_3_hi		

All statistic registers are 64 bits, and you must do two 32-bit reads. Intel recommends that you enable the `shadow_req[3:0]` in offset address 0x108 explained in *rsfec_debug_cfg* before reading the statistics register and disable after reading it.

Related Information

rsfec_debug_cfg on page 256

9.5.1. rsfec_top_clk_cfg

Description	Address	Addressing Mode
RS-FEC Clock configuration register	0x4	32-bits

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
11:8	fec_lane_ena	**Rsfec Clock/Lane Enable**	RW RO -	0xF

continued...

Bit	Name	Description	SW Access HW Access Protection	Reset
		Setting these bits enables the clock/lane for RS-FEC mode. This also enables the RX path to the RS-FEC core for that lane (else 0 the valid/data). One bit per lane [bit0=lane0]. If all lanes are disabled, the fec_clk are turned off. One of these bits MUST be set in order to properly access the registers which are in the fec_clk domain (rsfec_cfgcsr_core_csr).		
2:0	rsfec_clk_sel	**Clock selection for RS-FEC** Indicates which clock to use for rsfec core clock. In addition one of the fec_lane_ena bits must be set for the clock to propagate. 3'b000 : Select Ehip clock 3'b100 : Select EMIB Adapter TX clock 0 3'b101 : Select EMIB Adapter TX clock 1 3'b110 : Select EMIB Adapter TX clock 2 3'b111 : Select EMIB Adapter TX clock 3 All other inputs are invalid and defaults to Ehip clock	RW RO -	0x0

9.5.2. rsfec_top_tx_cfg

Description	Address	Addressing Mode
RS-FEC TX configuration register	0x10	32-bits

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:28	core_tx_pcs_bypass	**FEC TX Bypass** Setting this bit enables elane tx bypass to PMA interface. This is one bit per lane [bit0=lane0]	RW RO -	0x0
14:12	core_tx_in_sel3	**RS-FEC TX Select For Lane 3** Indicates which data to select for rsfec core TX input 3'b000 : Select EHIP Core TX Data - all lanes should have same selection 3'b001 : Select EHIP Lane TX Data 3'b010 : Select EMIB Lane TX Data with Deskew - should be same for all lanes 3'b011 : Select EMIB Lane TX Data No Deskew 3'b110 : FEC Lane Disabled - tie inputs to 0 3'b111 : Debug Mode - Select Loopback from output of RS-FEC RX	RW RO -	0x0
10:8	core_tx_in_sel2	**RS-FEC TX Select For Lane 2** Indicates which data to select for rsfec core TX input 3'b000 : Select EHIP Core TX Data - all lanes should have same selection 3'b001 : Select EHIP Lane TX Data 3'b010 : Select EMIB Lane TX Data with Deskew - should be same for all lanes 3'b011 : Select EMIB Lane TX Data No Deskew 3'b110 : FEC Lane Disabled - tie inputs to 0 3'b111 : Debug Mode - Select Loopback from output of RS-FEC RX	RW RO -	0x0
6:4	core_tx_in_sel1	**RS-FEC TX Select For Lane 1** Indicates which data to select for rsfec core TX input	RW RO -	0x0

continued...

Bit	Name	Description	SW Access HW Access Protection	Reset
		3'b000 : Select EHIP Core TX Data - all lanes should have same selection 3'b001 : Select EHIP Lane TX Data 3'b010 : Select EMIB Lane TX Data with Deskew - should be same for all lanes 3'b011 : Select EMIB Lane TX Data No Deskew 3'b110 : FEC Lane Disabled - tie inputs to 0 3'b111 : Debug Mode - Select Loopback from output of RS-FEC RX		
2:0	core_tx_in_sel0	**RS-FEC TX Select For Lane 0** Indicates which data to select for rsfec core TX input 3'b000 : Select EHIP Core TX Data - all lanes should have same selection 3'b001 : Select EHIP Lane TX Data 3'b010 : Select EMIB Lane TX Data with Deskew - should be same for all lanes 3'b011 : Select EMIB Lane TX Data No Deskew 3'b110 : FEC Lane Disabled - tie inputs to 0 3'b111 : Debug Mode - Select Loopback from output of RS-FEC RX	RW RO -	0x0

9.5.3. rsfec_top_rx_cfg

Description	Address	Addressing Mode
RS-FEC TX configuration register	0x14	32-bits

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:28	loopback_tx2rx	**FEC RX Bypass** Setting this bit enable loopback from TX RS-FEC to RX RS-FEC instead of getting from PMA interface. This is one bit per lane [bit0=lane0]	RW RO -	0x0
13:12	core_rx_out_sel3	**RS-FEC RX Output Select For Lane 3** Indicates which data to select for rsfec core TX input. All lanes should have same selection for EHIP and RS-FEC_DIRECT_100G modes. Does not work for TX Select of Elane and Loopback 2'b00 : Bypass RS-FEC RX paths - data from PMA interface fec / pcs path (normal bypass) 2'b01 : Select output of RS-FEC RX. 2'b10 : Bypass RS-FEC RX paths - data from PMA interface fec path for both ehip and elane 2'b11 : Debug Mode - Select Loopback from EHIP TX Data.	RW RO -	0x0
9:8	core_rx_out_sel2	**RS-FEC RX Output Select For Lane 2** Indicates which data to select for rsfec core TX input. All lanes should have same selection for EHIP and RS-FEC_DIRECT_100G modes. Does not work for TX Select of Elane and Loopback 2'b00 : Bypass RS-FEC RX paths - data from PMA interface fec / pcs path (normal bypass) 2'b01 : Select output of RS-FEC RX. 2'b10 : Bypass RS-FEC RX paths - data from PMA interface fec path for both ehip and elane	RW RO -	0x0

continued...

Bit	Name	Description	SW Access HW Access Protection	Reset
		2'b11 : Debug Mode - Select Loopback from EHIP TX Data.		
5:4	core_rx_out_sel1	**RS-FEC RX Output Select For Lane 1** Indicates which data to select for rsfec core TX input. All lanes should have same selection for EHIP and RS-FEC_DIRECT_100G modes. Does not work for TX Select of Elane and Loopback 2'b00 : Bypass RS-FEC RX paths - data from PMA interface fec / pcs path (normal bypass) 2'b01 : Select output of RS-FEC RX. 2'b10 : Bypass RS-FEC RX paths - data from PMA interface fec path for both ehip and elane 2'b11 : Debug Mode - Select Loopback from EHIP TX Data.	RW RO -	0x0
1:0	core_rx_out_sel0	**RS-FEC RX Output Select For Lane 0** Indicates which data to select for rsfec core TX input. All lanes should have same selection for EHIP and RS-FEC_DIRECT_100G modes. Does not work for TX Select of Elane and Loopback 2'b00 : Bypass RS-FEC RX paths - data from PMA interface fec / pcs path (normal bypass) 2'b01 : Select output of RS-FEC RX. 2'b10 : Bypass RS-FEC RX paths - data from PMA interface fec path for both ehip and elane 2'b11 : Debug Mode - Select Loopback from EHIP TX Data.	RW RO -	0x0

9.5.4. tx_aib_dsk_conf

Description	Address	Addressing Mode
Defines the configuration fields for TX Deskew	0x20	32-bits

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
7	tx_deskew_clear	**EMIB Deskew clear** 0:Normal deskew operation 1:TX EMIB deskew circuit in reset	RW RO -	0x0
3:0	tx_deskew_chan_sel	**Specifies which channels to include in the deskew procedure** 1= include	RW RO -	0x0

9.5.5. rsfec_core_cfg

Description	Address	Addressing Mode
RS-FEC core configuration	0x30	32-bits

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
1:0	frac	Main operation mode: 0 none: Non-fractured, supporting e.g. 100GE, 128GFC: One client using all 4 physical lanes. Register tables indexed by physical lane# only uses entry 0, unless otherwise specified. 1 reserved1 2 reserved2 3 frac4: Fractured, supporting e.g. 25GE, 32GFC: Four clients, each using 1 physical lane.	RW RO -	0x0

9.5.6. rsfec_lane_cfg

Register Name	Description	Address	Addressing Mode
rsfec_lane_cfg_0	RS-FEC per lane configuration	0x40	32-bits
rsfec_lane_cfg_1		0x44	
rsfec_lane_cfg_2		0x48	
rsfec_lane_cfg_3		0x4C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
3	rs544	Selects the RS encoder/decoder mode: 0: Use RS(528,514). 1: Use RS(544,514).	RW RO -	0x0
2	indic_byp	Bypass error indication (to reduce latency): 0: Sync headers in the 66b words extracted from uncorrectable FEC codewords are deliberately invalidated. 1: 66b words extracted from uncorrectable codewords are not explicitly marked bad. When number of symbol errors in a block of 8192 consecutive codewords has exceeded 417 with RS528 and 6380 with RS544, then sync header errors are generated towards the PCS layer for a period of 60ms to 75ms.	RW RO -	0x0
1	scr	Set to enable PN-5280 scrambling/descrambling. Must be set to 1 when RS-FEC_CORE_CFG.frac = frac4 and RS-FEC_LANE_CFG.fc = 1 (i.e. 32GFC), otherwise it must be set to 0.	RW RO -	0x0
0	fc	Set to enable Fibre Channel mode.	RW RO -	0x0

9.5.7. tx_aib_dsk_status

Description	Address	Addressing Mode
Status fields for TX Deskew	0x104	32-bits

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
11:8	tx_dsk_active_chans	**Active Channels.** 1 bit per channel (bit0 = EMIB chan 0). Indicates which EMIB channels received a deskew marker (since reset). This is a sticky bit that clears on reset and i_dsk_clear. For logging in RS-FEC CSR. CSR may treat this as a real time status signal.	RO WO -	0x0
7:4	tx_dsk_monitor_err	**Skew Monitor Error Detected** 1 bit per channel (bit0 =EMIB chan 0). This field only updates after a successful deskew. A non-zero value indicates that the deskewed Markers are no longer in alignment. 1 in each bit position indicates that the corresponding channel received a Deskew Marker before ALL enabled-channels received them. Note: You can see which channels received Deskew Markers (at all) via o_tx_dsk_active_chans.	RO WO -	0x0
3:1	tx_dsk_status	**Total-Channel-to-channel-skew status** Valid when tx_dsk_eval_done=1. Reports the total skew detected at the end of the deskew procedure 0..5: 0..5 cycles of delay added to remove skew 6..7:error detected	RO WO -	0x0
0	tx_dsk_eval_done	**Deskew Complete** Means Deskew procedure has completed	RO WO -	0x0

9.5.8. rsfec_debug_cfg

Description	Address	Addressing Mode
Extra config/debug on fec_clock	0x108	32-bits

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31	main_rst	**Main Soft Reset** Setting this bit causes main soft reset of RS-FEC including tx/rx path	RW RO -	0x0
29	rx_rst	**RX Soft Reset** Setting this bit causes soft reset of the RX datapath in RS-FEC Core	RW RO -	0x0
28	tx_rst	**TX Soft Reset** Setting this bit causes soft reset of the TX datapath in RS-FEC Core	RW RO -	0x0
7:4	shadow_clear	**Clear Rsfec Counters** 1: Clear the collection and shadow counters so that the next shadow request or snapshot starts from 0. If the counters are not cleared, they continue counting and rollover.	RW RO -	0x0
3:0	shadow_req		RW RO -	0x0

9.5.9. rsfec_lane_tx_stat

Register Name	Description	Address	Addressing Mode
rsfec_lane_tx_stat_0	RS-FEC per lane TX status	0x120	32-bits
rsfec_lane_tx_stat_1		0x124	
rsfec_lane_tx_stat_2		0x128	
rsfec_lane_tx_stat_3		0x12C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
3	pace_inv	PCS TX pacing violation. With RS528 .pace_inv is never set. With RS544 .pace_inv is set when the layer above presents TX data in more than 33 consecutive cycles.	RO WO -	0x0
2	resync	PCS TX alignment/codeword marker resync. Not valid when RS-FEC_LANE_CFG1.eng_cust_am_en = 1.	RO WO -	0x0
1	blk_inv	PCS TX 66b invalid block type. Not valid when transcoding is bypassed.	RO WO -	0x0
0	hdr_inv	PCS TX 66b invalid sync header. Not valid when transcoding is bypassed.	RO WO -	0x0

9.5.10. rsfec_lane_tx_hold

Register Name	Description	Address	Addressing Mode
rsfec_lane_tx_hold_0	RS-FEC per lane TX status hold	0x130	32-bits
rsfec_lane_tx_hold_1		0x134	
rsfec_lane_tx_hold_2		0x138	
rsfec_lane_tx_hold_3		0x13C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
3	pace_inv	PCS TX pacing violation. With RS528 .pace_inv is never set. With RS544 .pace_inv is set when the layer above presents TX data in more than 33 consecutive cycles.	W1C W1S -	0x0
2	resync	PCS Tx alignment/codeword marker resync. Not valid when RSFEC_LANE_CFG1.eng_cust_am_en = 1.	W1C W1S -	0x0
1	blk_inv	PCS Tx 66b invalid block type.	W1C	0x0

continued...

Bit	Name	Description	SW Access HW Access Protection	Reset
		Not valid when transcoding is bypassed.	W1S -	
0	hdr_inv	PCS Tx 66b invalid sync header. Not valid when transcoding is bypassed.	W1C W1S -	0x0

9.5.11. rsfec_lane_tx_inten

Register Name	Description	Address	Addressing Mode
rsfec_lane_tx_inten_0	RS-FEC per lane TX status hold interrupt - set to 1 to enable rsfec_lane_tx lane interrupt	0x140	32-bits
rsfec_lane_tx_inten_1		0x144	
rsfec_lane_tx_inten_2		0x148	
rsfec_lane_tx_inten_3		0x14C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
3	pace_inv	PCS TX pacing violation. When RSFEC_LANE_CFG.rs544 = 0 .pace_inv is never set. When RSFEC_LANE_CFG.rs544 = 1 .pace_inv is set when there is more than 33 consecutive non-idle cycles.	RW RO -	0x0
2	resync	PCS TX alignment/codeword marker resync. Not valid when RSFEC_LANE_CFG1.eng_cust_am_en = 1.	RW RO -	0x0
1	blk_inv	PCS TX 66b invalid block type. Not valid when transcoding is bypassed.	RW RO -	0x0
0	hdr_inv	PCS TX 66b invalid sync header. Not valid when transcoding is bypassed.	RW RO -	0x0

9.5.12. rsfec_lane_rx_stat

Register Name	Description	Address	Addressing Mode
rsfec_lane_rx_stat_0	RS-FEC per lane RX status	0x150	32-bits
rsfec_lane_rx_stat_1		0x154	
rsfec_lane_rx_stat_2		0x158	
rsfec_lane_rx_stat_3		0x15C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
6	uncorr_cw	Set when a FEC code word was not corrected due to too many errors.	RO WO -	0x0
5	corr_cw	Set when a FEC code word had one or more errors that were corrected.	RO WO -	0x0
4	hi_ser	High symbol error rate. Set when the number of symbol errors in a window of 8192 consecutive codewords has exceeded 417 with RS528 and 6380 with RS544. If RSFEC_LANE_CFG.indic_byp = 1, then sync header errors are generated towards the PCS layer for a period of 60ms to 75ms.	RO WO -	0x0
3	am_5bad	RX was locked (and aligned if RSFEC_CORE_CFG.frac = none) but 5 consecutive alignment/codeword markers were not valid. Restarts the synchronization.	RO WO -	0x0
2	fec_3bad	RX was locked (and aligned if RSFEC_CORE_CFG.frac = none) but 3 consecutive FEC codewords wer not corrected. Restarts the synchronization.	RO WO -	0x0
1	not_locked	RX lane not locked. Not locked to alignment/codeword markers (100GE/128GFC/25GE) or to FEC codewords (32GFC). One entry per physical lane, regardless of RSFEC_CORE_CFG.frac.	RO WO -	0x0
0	sf	Incoming signal fail (transceiver unable to lock to signal). One entry per physical lane, regardless of RSFEC_CORE_CFG.frac.	RO WO -	0x0

9.5.13. rsfec_lane_rx_hold

Register Name	Description	Address	Addressing Mode
rsfec_lane_rx_hold_0	RS-FEC per lane RX status hold	0x160	32-bits
rsfec_lane_rx_hold_1		0x164	
rsfec_lane_rx_hold_2		0x168	
rsfec_lane_rx_hold_3		0x16C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
6	uncorr_cw	Set when a FEC code word was not corrected due to too many errors.	W1C W1S -	0x0
5	corr_cw	Set when a FEC code word had one or more errors that were corrected.	W1C W1S -	0x0
4	hi_ser	High symbol error rate.	W1C	0x0

continued...

Bit	Name	Description	SW Access HW Access Protection	Reset
		Set when the number of symbol errors in a window of 8192 consecutive codewords has exceeded 417 with RS528 and 6380 with RS544. If RSFEC_LANE_CFG.indic_byp = 1, then sync header errors are generated towards the PCS layer for a period of 60ms to 75ms.	W1S -	
3	am_5bad	RX was locked (and aligned if RSFEC_CORE_CFG.frac = none) but 5 consecutive alignment/codeword markers were not valid. Restarts the synchronization.	W1C W1S -	0x0
2	fec_3bad	RX was locked (and aligned if RSFEC_CORE_CFG.frac = none) but 3 consecutive FEC codewords were not corrected. Restarts the synchronization.	W1C W1S -	0x0
1	not_locked	RX lane not locked. Not locked to alignment/codeword markers (100GE/128GFC/25GE) or to FEC codewords (32GFC). One entry per physical lane, regardless of RSFEC_CORE_CFG.frac.	W1C W1S -	0x0
0	sf	Incoming signal fail (transceiver unable to lock to signal). One entry per physical lane, regardless of RSFEC_CORE_CFG.frac.	W1C W1S -	0x0

9.5.14. rsfec_lane_rx_inten

Register Name	Description	Address	Addressing Mode
rsfec_lane_rx_inten_0	RS-FEC per lane RX status hold interrupt - set to 1 to enable rsfec_lane_rx lane interrupt	0x170	32-bits
rsfec_lane_rx_inten_1		0x174	
rsfec_lane_rx_inten_2		0x178	
rsfec_lane_rx_inten_3		0x17C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
6	uncorr_cw	Set when a FEC code word was not corrected due to too many errors.	RW RO -	0x0
5	corr_cw	Set when a FEC code word had one or more errors that were corrected.	RW RO -	0x0
4	hi_ser	High symbol error rate. Set when the number of symbol errors in a window of 8192 consecutive codewords has exceeded 417 with RS528 and 6380 with RS544. If RSFEC_LANE_CFG.indic_byp = 1, then sync header errors are generated towards the PCS layer for a period of 60ms to 75ms.	RW RO -	0x0
3	am_5bad	RX was locked (and aligned if RSFEC_CORE_CFG.frac = none) but 5 consecutive alignment/codeword markers were not valid.	RW RO	0x0

continued...

Bit	Name	Description	SW Access HW Access Protection	Reset
		Restarts the synchronization.	-	
2	fec_3bad	RX was locked (and aligned if RSFEC_CORE_CFG.frac = none) but 3 consecutive FEC codewords were not corrected. Restarts the synchronization.	RW RO -	0x0
1	not_locked	RX lane not locked. Not locked to alignment/codeword markers (100GE/128GFC/25GE) or to FEC codewords (32GFC). One entry per physical lane, regardless of RSFEC_CORE_CFG.frac.	RW RO -	0x0
0	sf	Incoming signal fail (transceiver unable to lock to signal). One entry per physical lane, regardless of RSFEC_CORE_CFG.frac.	RW RO -	0x0

9.5.15. rsfec_lanes_rx_stat

Description	Address	Addressing Mode
RS-FEC combined lanes RX status	0x180	32-bits

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
1	not_deskew	All RX lanes locked but the alignment markers were not unique or the skew was too large. This is an event signal, so use .not_align above instead to determine the alignment state. Restarts the synchronization. Only applicable when RSFEC_CORE_CFG.frac = none (100GE/128GFC).	RO WO -	0x0
0	not_align	RX lanes not aligned (state). Incoming signal fail, RX lanes not all locked, alignment markers not unique or skew too large. Only applicable when RSFEC_CORE_CFG.frac = none (100GE/128GFC).	RO WO -	0x0

9.5.16. rsfec_lanes_rx_hold

Description	Address	Addressing Mode
RS-FEC combined lanes RX hold status	0x188	32-bits

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
1	not_deskew	All RX lanes locked but the alignment markers were not unique or the skew was too large. This is an event signal, so use .not_align above instead to determine the alignment state. Restarts the synchronization.	W1C W1S -	0x0

continued...

Bit	Name	Description	SW Access HW Access Protection	Reset
		Only applicable when RSFEC_CORE_CFG.frac = none (100GE/128GFC).		
0	not_align	RX lanes not aligned (state). Incoming signal fail, RX lanes not all locked, alignment markers not unique or skew too large. Only applicable when RSFEC_CORE_CFG.frac = none (100GE/128GFC).	W1C W1S -	0x0

9.5.17. rsfec_lanes_rx_inten

Description	Address	Addressing Mode
RS-FEC combined lanes RX interrupt enable - set to 1 to enable rsfec_lanes rx lane interrupt	0x18C	32-bits

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
1	not_deskew	All RX lanes locked but the alignment markers were not unique or the skew was too large. This is an event signal, so use .not_align above instead to determine the alignment state. Restarts the synchronization. Only applicable when RSFEC_CORE_CFG.frac = none (100GE/128GFC).	RW RO -	0x0
0	not_align	RX lanes not aligned (state). Incoming signal fail, RX lanes not all locked, alignment markers not unique or skew too large. Only applicable when RSFEC_CORE_CFG.frac = none (100GE/128GFC).	RW RO -	0x0

9.5.18. rsfec_ln_mapping_rx

Register Name	Description	Address	Addressing Mode
rsfec_ln_mapping_rx_0	RS-FEC FEC lane mapping	0x1A0	32-bits
rsfec_ln_mapping_rx_1		0x1A4	
rsfec_ln_mapping_rx_2		0x1AB	
rsfec_ln_mapping_rx_3		0x1AC	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
1:0	fec_lane	FEC lane# received on each physical lane. Only applicable when RSFEC_CORE_CFG.frac = none (100GE/128GFC).	RO WO -	0x0

9.5.19. rsfec_ln_skew_rx

Register Name	Description	Address	Addressing Mode
rsfec_ln_skew_rx_0	RS-FEC FEC lane skew	0x1B0	32-bits
rsfec_ln_skew_rx_1		0x1B4	
rsfec_ln_skew_rx_2		0x1B8	
rsfec_ln_skew_rx_3		0x1BC	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
6:0	skew	Lane skew value (unit is 80 bits). Only valid when the RX lanes are aligned. Only applicable when RSFEC_CORE_CFG.frac = none (100GE/128GFC).	RO WO -	0x00

9.5.20. rsfec_cw_pos_rx

Register Name	Description	Address	Addressing Mode
rsfec_cw_pos_rx_0	RS-FEC codeword bit position on RX	0x1C0	32-bits
rsfec_cw_pos_rx_1		0x1C4	
rsfec_cw_pos_rx_2		0x1C8	
rsfec_cw_pos_rx_3		0x1CC	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
12:0	num	Bit number of first bit in FEC codeword. Only intended for debug of deterministic latency.	RO WO -	0x0000

9.5.21. rsfec_core_ecc_hold

Description	Address	Addressing Mode
RS-FEC SRAM ECC status hold	0x1D0	32-bits

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
15:8	mbe	SRAM ECC uncorrectable error detected. Same bit ordering as for .sbe above.	W1C WO1S -	0x00
7:0	sbe	SRAM ECC correctable (single bit) error detected.	W1C	0x00

continued...

Bit	Name	Description	SW Access HW Access Protection	Reset
		Should not become set. One bit per SRAM. Bits 0-3 covers the deskew buffers for physical lanes 0-3 (only used when RSFEC_CORE_CFG.frac = none). Bits 4-7 covers the data buffers for the RS decoding. When RSFEC_CORE_CFG.frac = frac4, these are used 1:1 for the physical lanes 0-3.	WO1S -	

9.5.22. rsfec_err_inj_tx

Register Name	Description	Address	Addressing Mode
rsfec_err_inj_tx_0	RS-FEC error injection mode	0x1E0	32-bits
rsfec_err_inj_tx_1		0x1E4	
rsfec_err_inj_tx_2		0x1E8	
rsfec_err_inj_tx_3		0x1EC	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
15:8	pat	TX error injection pattern for each lane. Value specifies which bits are being toggled on each lane, when that lane is hit. There is an 8b pattern per lane. When a 66b word on a lane is hit, 8 consecutive bits out of these 66 are XOR'ed with the pattern. One entry per physical lane, regardless of RSFEC_CORE_CFG.frac.	RW RO -	0x00
7:0	rate	TX error injection rate for each physical lane. Data is output towards the PMA 66 bits at a time (not to be confused with 66b PCS symbols). The value specifies the fraction of such 66b words to hit. The unit is 1/256th so a value of, say, 7 causes 7/256th of the 66b words being sent on the lane to be hit. One entry per physical lane, regardless of RSFEC_CORE_CFG.frac.	RW RO -	0x00

9.5.23. rsfec_err_val_tx

Register Name	Description	Address	Addressing Mode
rsfec_err_val_tx_0	RS-FEC per lane error injection status	0x1F0	32-bits
rsfec_err_val_tx_1		0x1F4	
rsfec_err_val_tx_2		0x1F8	
rsfec_err_val_tx_3		0x1FC	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
15:8	inj1s	Same for bits changed from 0 to 1 on each physical lane.	RO WO	0x00

continued...

Bit	Name	Description	SW Access HW Access Protection	Reset
			-	
7:0	inj0s	Number of bits (modulo 256) that were changed from 1 to 0 on each physical lane. Cleared when the corresponding RSFEC_ERR_INJ_TX.rate is written with a non-zero value after being all zero, i.e. when a test is initiated. A value read from this register is not reliable while injecting. A value read is reliable when the test is completed, i.e. after the lane's RSFEC_ERR_INJ_TX.rate or RSFEC_ERR_INJ_TX.pat has been cleared. One entry per physical lane, regardless of RSFEC_CORE_CFG.frac.	RO WO -	0x00

9.5.24. rsfec_corr_cw_cnt (Low)

Register Name	Description	Address	Addressing Mode
rsfec_corr_cw_cnt_0_lo	RS-FEC number of FEC codewords with errors that were corrected (low word: bits 31 to 0)	0x200	32-bits
rsfec_corr_cw_cnt_1_lo		0x208	
rsfec_corr_cw_cnt_2_lo		0x210	
rsfec_corr_cw_cnt_3_lo		0x218	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:0	stat	Statistics value.	RO WO -	0x0000 0000

9.5.25. rsfec_corr_cw_cnt (High)

	Description	Address	Addressing Mode
rsfec_corr_cw_cnt_0_hi	RS-FEC number of FEC codewords with errors that were corrected (high word: bits 63 to 32)	0x204	32-bits
rsfec_corr_cw_cnt_1_hi		0x20C	
rsfec_corr_cw_cnt_2_hi		0x214	
rsfec_corr_cw_cnt_3_hi		0x21C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:0	stat	Statistics value.	RO WO -	0x0000 0000

9.5.26. rsfec_uncorr_cw_cnt (Low)

Register Name	Description	Address	Addressing Mode
rsfec_uncorr_cw_cnt_0_lo	RS-FEC number of FEC codewords that could not be corrected due to too many errors (low word: bits 31 to 0)	0x220	32-bits
rsfec_uncorr_cw_cnt_1_lo		0x228	
rsfec_uncorr_cw_cnt_2_lo		0x230	
rsfec_uncorr_cw_cnt_3_lo		0x238	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:0	stat	Statistics value.	RO WO -	0x0000 0000

9.5.27. rsfec_uncorr_cw_cnt (High)

Register Name	Description	Address	Addressing Mode
rsfec_uncorr_cw_cnt_0_hi	RS-FEC number of FEC codewords that could not be corrected due to too many errors (high word: bits 63 to 32)	0x224	32-bits
rsfec_uncorr_cw_cnt_1_hi		0x22C	
rsfec_uncorr_cw_cnt_2_hi		0x234	
rsfec_uncorr_cw_cnt_3_hi		0x23C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:0	stat	Statistics value.	RO WO -	0x0000 0000

9.5.28. rsfec_corr_syms_cnt (Low)

Register Name	Description	Address	Addressing Mode
rsfec_corr_syms_cnt_0_lo	RS-FEC number of 10b symbols corrected for the lane (low word: bits 31 to 0)	0x240	32-bits
rsfec_corr_syms_cnt_1_lo		0x248	
rsfec_corr_syms_cnt_2_lo		0x250	
rsfec_corr_syms_cnt_3_lo		0x258	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:0	stat	Statistics value.	RO	0x0000 0000

Bit	Name	Description	SW Access HW Access Protection	Reset
			WO -	

9.5.29. rsfec_corr_syms_cnt (High)

Register Name	Description	Address	Addressing Mode
rsfec_corr_syms_cnt_0_hi	RS-FEC number of 10b symbols corrected for the lane (high word: bits 63 to 32)	0x244	32-bits
rsfec_corr_syms_cnt_1_hi		0x24C	
rsfec_corr_syms_cnt_2_hi		0x254	
rsfec_corr_syms_cnt_3_hi		0x25C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:0	stat	Statistics value.	RO WO -	0x0000 0000

9.5.30. rsfec_corr_0s_cnt (Low)

Register Name	Description	Address	Addressing Mode
rsfec_corr_0s_cnt_0_lo	RS-FEC number of bits corrected 0->1 for the lane (low word: bits 31 to 0)	0x260	32-bits
rsfec_corr_0s_cnt_1_lo		0x268	
rsfec_corr_0s_cnt_2_lo		0x270	
rsfec_corr_0s_cnt_3_lo		0x278	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:0	stat	Statistics value.	RO WO -	0x0000 0000

9.5.31. rsfec_corr_0s_cnt (High)

Register Name	Description	Address	Addressing Mode
rsfec_corr_0s_cnt_0_hi	RS-FEC number of bits corrected 0->1 for the lane (high word: bits 63 to 32)	0x264	32-bits
rsfec_corr_0s_cnt_1_hi		0x26C	
rsfec_corr_0s_cnt_2_hi		0x274	
rsfec_corr_0s_cnt_3_hi		0x27C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:0	stat	Statistics value.	RO WO -	0x0000 0000

9.5.32. rsfec_corr_1s_cnt (Low)

Register Name	Description	Address	Addressing Mode
rsfec_corr_1s_cnt_0_lo	RS-FEC number of bits corrected 1->0 for the lane (low word: bits 31 to 0)	0x280	32-bits
rsfec_corr_1s_cnt_1_lo		0x288	
rsfec_corr_1s_cnt_2_lo		0x290	
rsfec_corr_1s_cnt_3_lo		0x298	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:0	stat	Statistics value.	RO WO -	0x0000 0000

9.5.33. rsfec_corr_1s_cnt (High)

Register Name	Description	Address	Addressing Mode
rsfec_corr_1s_cnt_0_hi	RS-FEC number of bits corrected 1->0 for the lane (high word: bits 63 to 32)	0x284	32-bits
rsfec_corr_1s_cnt_1_hi		0x288	
rsfec_corr_1s_cnt_2_hi		0x294	
rsfec_corr_1s_cnt_3_hi		0x29C	

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	SW Access HW Access Protection	Reset
31:0	stat	Statistics value.	RO	0x0000 0000

Bit	Name	Description	SW Access HW Access Protection	Reset
			WO -	

9.6. Register Map Revision History

Document Version	Changes
2020.01.31	Made the following changes: • Clarified the swizzle settings for *0x0013: TX/RX Polarity and Gray Code Encoding*. • Added *Enabling and Disabling Electrical Idle Detector Filtering and Reading Electrical Idle Detector Status*. • Added *Initial Adaptation Effort Levels*. • Added *Load a PMA Configuration*.
2019.10.11	Made the following changes: • Added the "Loading PMA Configuration Register LOW_POWER_MODE" figure. • In 0x002B, removed 0x84[0]: 1'b0 and changed VCC to VCCH_GXE. • Updated the odd eye values in *Reading NRZ/PAM4 Eye Height*. • Added the "Supported Baud Rate Ratios" table. • Added *0x0020: Electrical Idle Detector*. • Clarified that RS-FEC register back-to-back writes are allowed.
2019.07.29	Made the following changes: • Added *Check the PMA Adaptation Status*. • Added the "RS-FEC AVMM Interface Ports" table and related information to *RS-FEC Registers*. • Added *0x0030: PMA Mux Clock Swap*. • Added *0x000E: RX Phase Slip*.
2019.04.19	Made the following changes: • Added "0x84[2:0]: 3'b110 to set to PRBS13" to *0x0002: PMA PRBS Settings*. • Added a reference to the "TX Equalization Settings for PAM4 and NRZ Signals" table in *TX Equalizer* to *0x0015: TX Equalization*. • Added the RF_A parameter to *Reading PMA Analog Parameters*. • Reading *NRZ/PAM4 Eye Height*.
2019.02.04	Made the following changes: • Split *PMA Capability Registers* into *PMA Capability Registers* and *PMA Control and Status Registers*. • Added *PMA Registers 0x200 to 0x203 Usage*. • Added registers 0x40143 and 0x40144. • Updated *Reading PMA Analog Parameters*, *Updating PMA Analog Parameters*, and *Loading Parameters into the Receiver*. • Added *Fixing Parameter Values*. • Updated some parameter names. • Changed the description in the "0x002B: RX Termination and TX Driver Tri-state Behavior" section.
2018.10.08	Made the following changes: • Removed the following addresses from the "PMA AVMM Registers" section: — 0xE8 — 0xE9 — 0xEA — 0xEB • Added the "0x002C: Read PMA Analog Parameter" section. • Added the "0x006C: Set the PMA Analog Parameter" section. • Added the "0x00EC: Load the PMA Analog Parameter to the PMA RX" section.

continued...

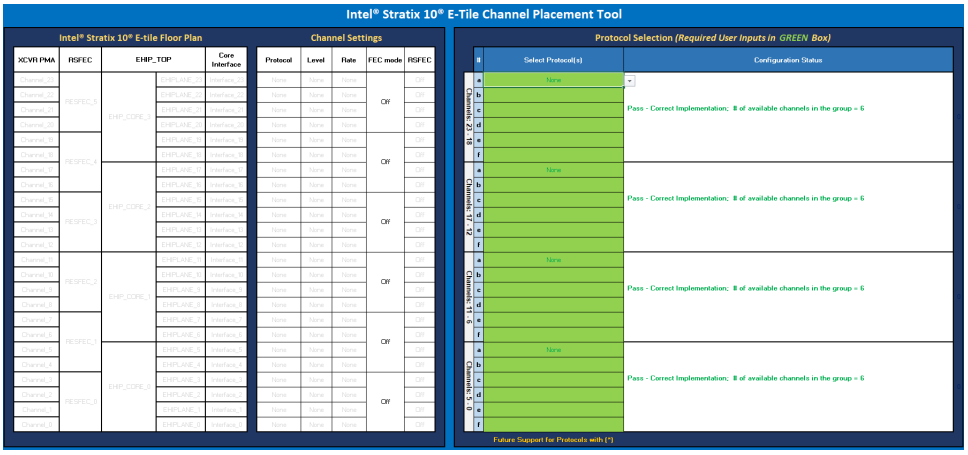
Document Version	Changes
	• Added the "RS-FEC Registers" section and all subsections. • Changed the description in the "0x000A: Receiver Tuning Controls" section. • Changed the description in the "0x0126: Read Receiver Tuning Parameters" section.
2018.07.18	Made the following changes: • Added the Name and Type columns and updated descriptions in the "PMA Capability Register Map" table. • Added new registers to the "PMA AVMM Registers" section. • Updated the description in the "0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control" section. • Updated the description in the "0x0018: Status/Debug Register" section.
2018.05.15	Made the following changes: • Changed the addresses in the "PMA/PCS Avalon-MM Register Map" table. • Changed the description of address 0x000A in the PMA Attribute Codes" table. • Added address 0x0011 to the "PMA Attribute Codes" table. • Changed the descriptions for addresses 0x0015 and 0x0018 in the "PMA Attribute Codes" table. • Added bit offsets [3:2] and [4] to address 0x9 in the "PMA Register Map" table. • Removed addresses 0x50040 and 0x50041 in the "PMA Capability Register Map" table. • Added address 0x8B to the "PMA Register Map" table.
2018.01.31	Initial release.

A. E-Tile Channel Placement Tool

E-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 *E-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.

The Excel-based *E-Tile Channel Placement Tool*, supplemented with **Instructions**, **Legend**, and **Revision** tabs, is available for download at E-Tile Channel Placement Tool.

Figure 120. E-Tile Channel Placement Tool



Related Information

- Intel Stratix 10 Device Family Pin Connection Guidelines
- Intel Agilex Device Family Pin Connection Guidelines

A.1. E-Tile Channel Placement Tool Revision History

Document Version	Changes
2020.01.31	Made the following change: • Added the serial lite IV NRZ protocol.
2019.10.11	Made the following changes:

continued...

Document Version	Changes
	• Added the Related Information links for the Intel Agilex device documents. • Added aggregate FEC modes for PMA direct, both PAM4 and NRZ. • Clarified that the tool is for E-tile in general. • Added Intel Stratix 10 DX Packages and Intel Agilex Packages tabs to the tool.
2019.07.29	Made the following changes: • Removed multi-channel selections from non-FEC channels to streamline the Protocol drop down list. • Revised the fractured FEC note to indicate that: — Only one IP can be instantiated per FEC block when using fractured FEC. — Unique protocols on one fractured FEC block can only be achieved using dynamic reconfiguration. • Added new multi-channel selections, mainly emphasizing those using fractured FEC or PTP due to the single IP per FEC block restriction when using those modes. • Changed PTP so that it can no longer be selected as a standalone.
2019.02.04	Made the following changes: • Added information for clocking fractured FEC. • Added new protocol: Channel PLL. • Updated PAM4 protocol formatting.
2018.05.15	Made the following changes: • Renamed the "Channel Use Model" appendix to "E-Tile Channel Placement Tool." • Removed PMA Direct Single Channel Mode, PMA Direct Dual Channel Mode, and Precision Time Protocol (PTP) Placement. • Added a description, screenshot and link to the "E-Tile Channel Placement Tool" and Intel Stratix 10 PCG.
2018.01.31	Initial release.

B. PMA Direct PAM4 30 Gbps to 57.8 Gbps Implementation

The PAM4 modulation scheme can help you achieve greater Ethernet speeds such as 200G/400G with efficiency and quality. Many protocol standards currently use the PAM4 coding scheme, such as 400GBASE-SR16, 200GBASE-KR4, and 100GBASE-CR2.

This chapter introduces a design example using a PMA direct PAM4 57.8 Gbps 12-channel design. The flow guides you through the following elements:

- Intel Quartus Prime Pro Edition IP instantiation
- Pin assignment
- Compilation
- Board bring up
- Debug tool introduction

B.1. Building Blocks and Considerations

Use the *E-Tile Channel Placement Tool* to configure 24 channels of your PMA direct PAM4 30 Gbps to 57.8 Gbps design.

Start by selecting your protocols from **Select Protocol(s)** column. 24 channels of PMA direct PAM4 30 Gbps to 57.8 Gbps designs use:

- 12 even PMA channels bonded out, all 24 channels instantiated
- 24 core interfaces

Figure 121. E-Tile Channel Placement Tool

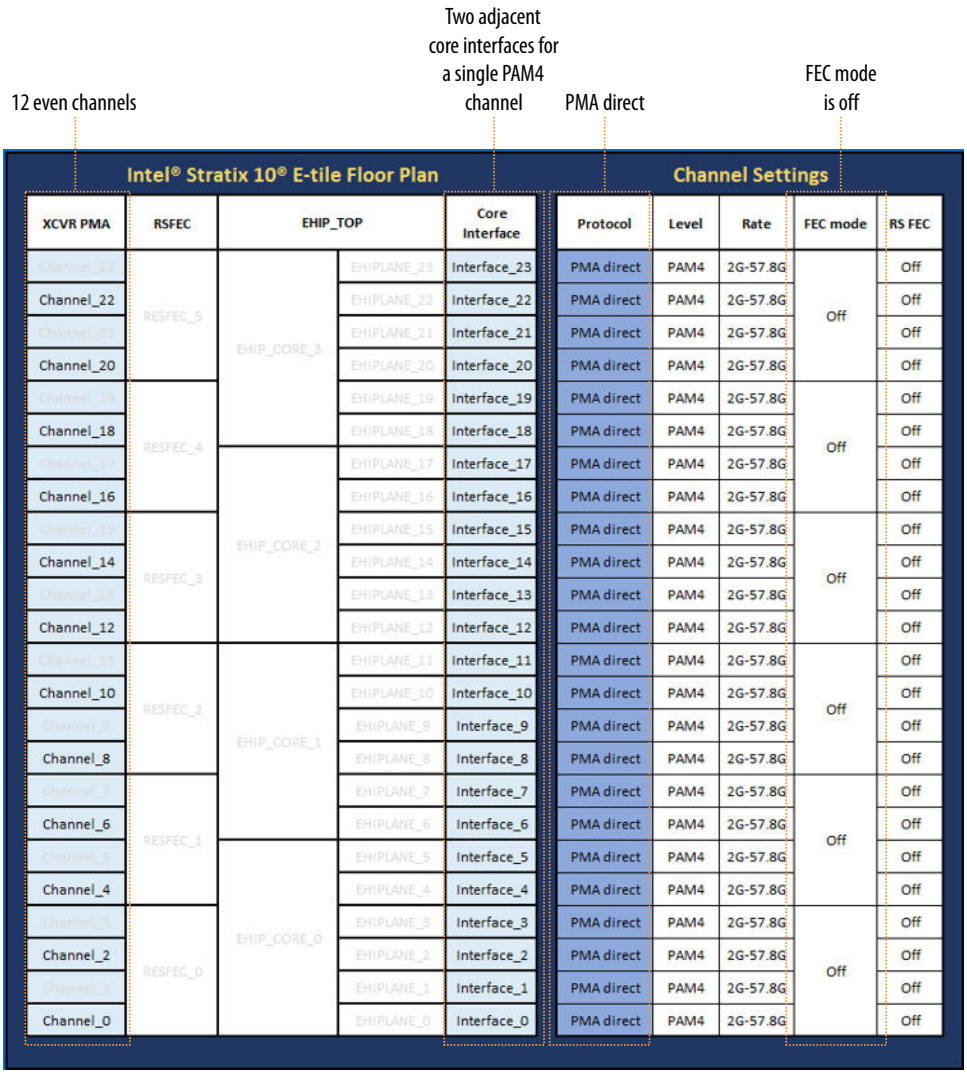


Figure 122. PMA Direct Datapath (Channels 0 to 11)

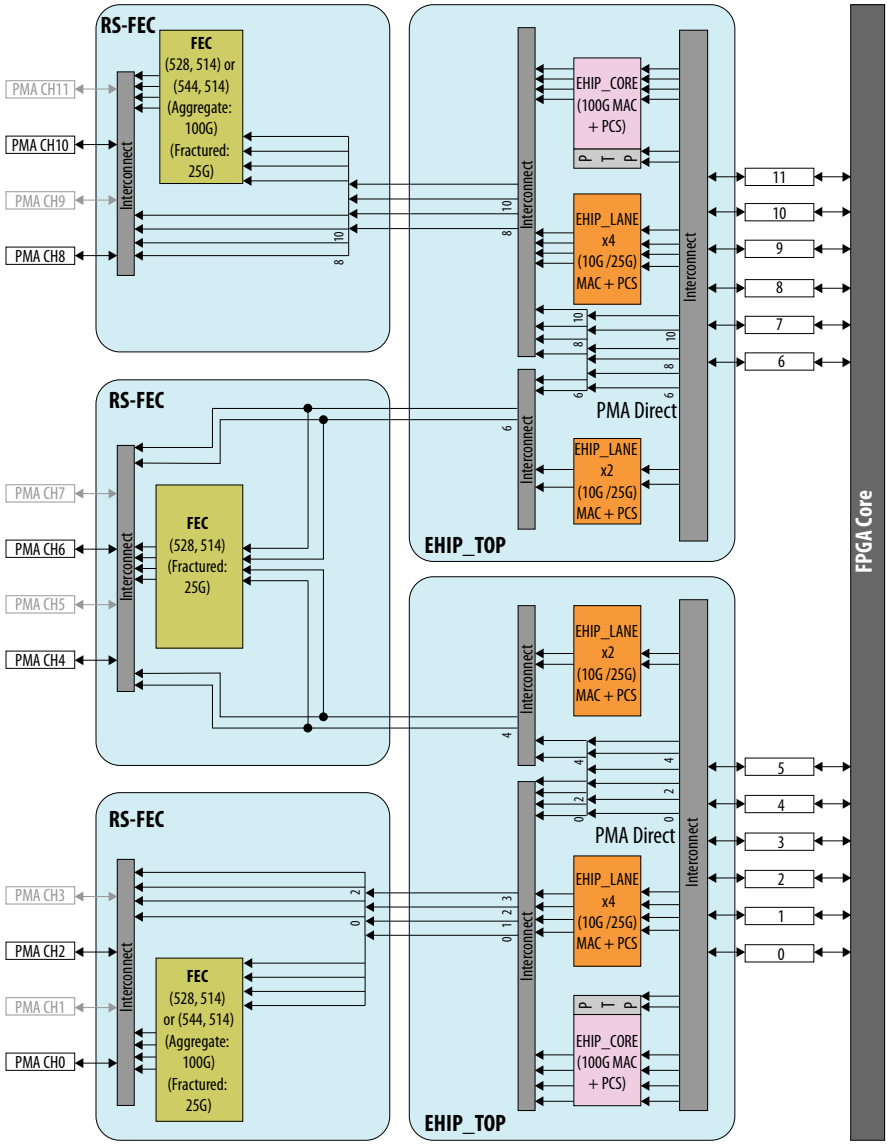


Figure 123. PMA Direct Datapath (Channels 12 to 23)

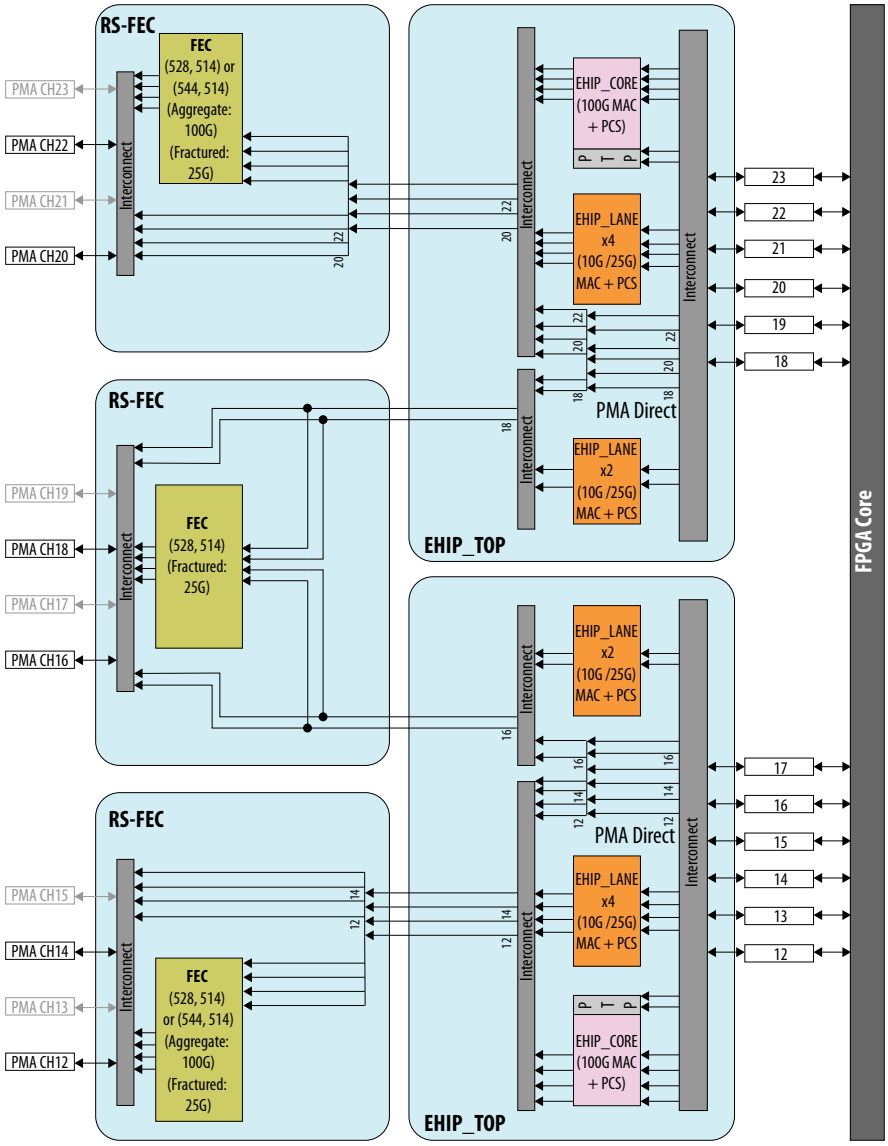
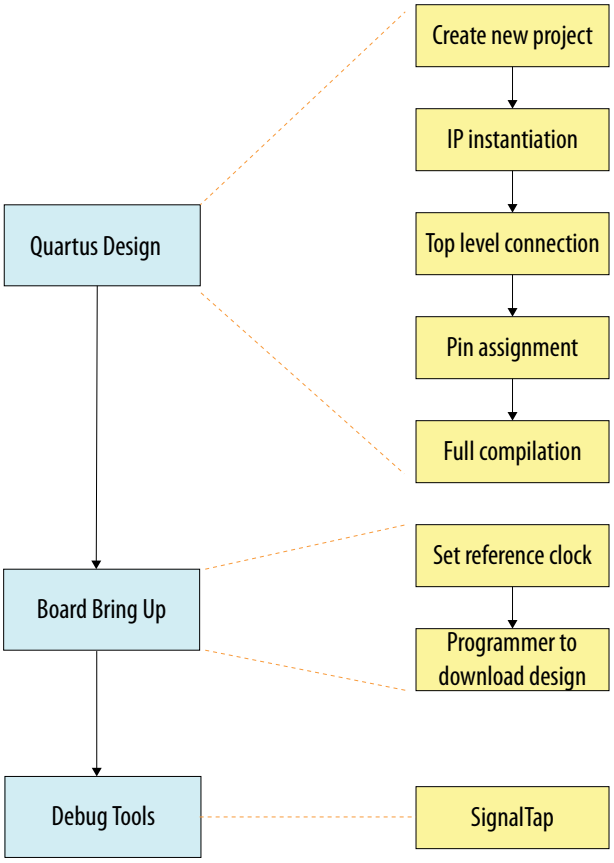


Figure 124. Design Flow Overview



Related Information

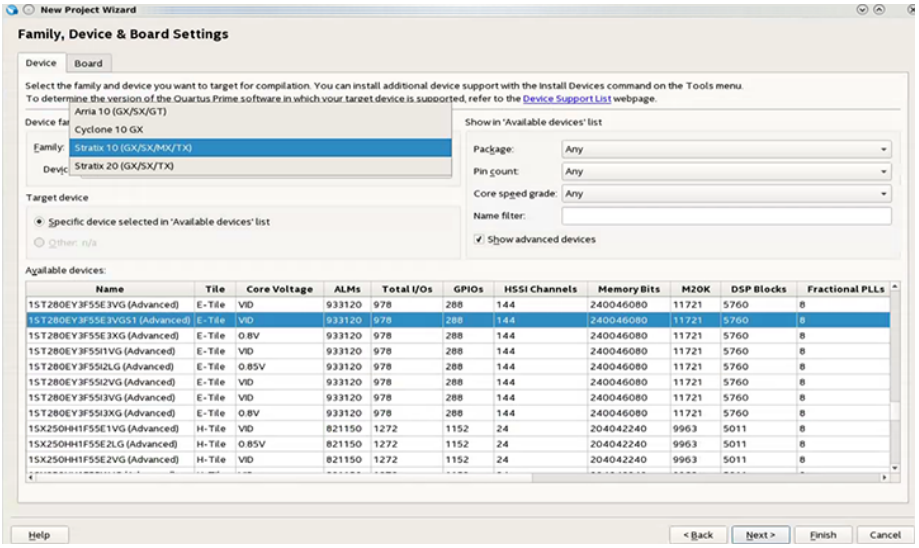
E-Tile Channel Placement Tool

B.2. Starting a New Intel Quartus Prime Pro Edition Design

This design example uses Intel Quartus Prime Pro Edition software version 18.0.

1. Click **File > New Project Wizard**.
2. Select a project folder, then keep clicking **Next** until you see **Family, Device & Board Settings**.
3. Select your device, then keep clicking **Next** until you reach the end of the project settings, then click **Finish**.

Figure 125. Family, Device & Board Settings



Refer to the *Device Data Sheet* for E-tile specifications.

Related Information

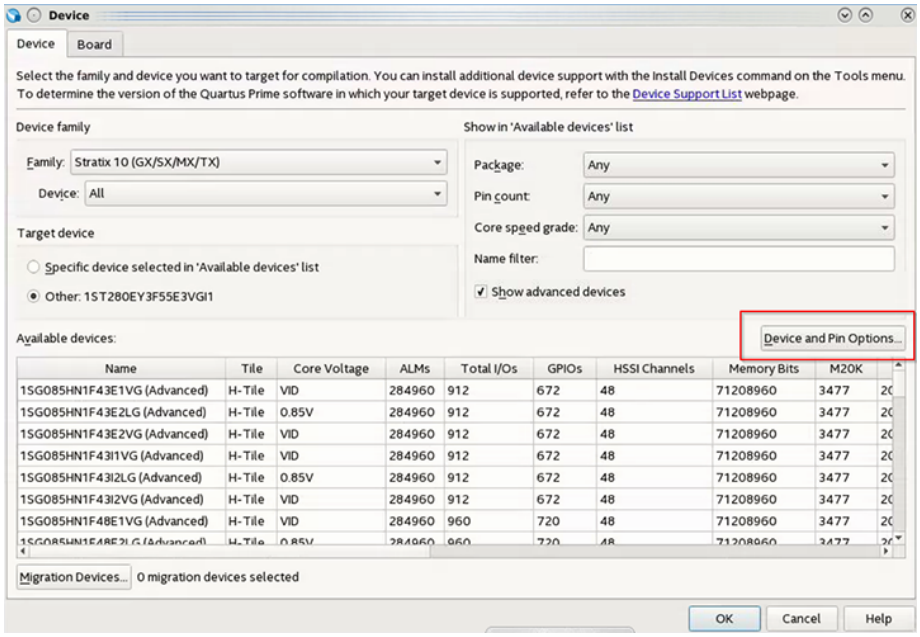
- Intel Stratix 10 Device Data Sheet
- Intel Agilex Device Data Sheet

B.3. Selecting the Configuration Clock Source

Use this procedure to set the clock for the transceiver reset sequence (TRS) and local TRS (LTRS) blocks.

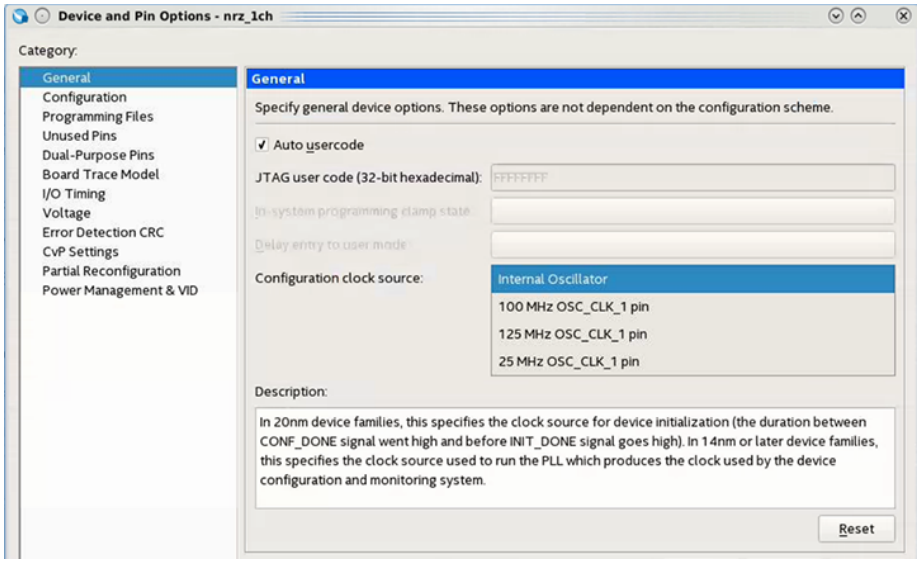
1. Click **Assignment > Settings > Device/Board.**
2. Click the **Device and Pin Options** button.

Figure 126. Device and Pin Options Button



3. From the **General** category, select either **100 MHz OSC_CLK_1 pin**, **125 MHz OSC_CLK_1 pin**, or **25 MHz OSC_CLK_1 pin** in the **Configuration clock source** field depending on your clock frequency's availability.

Figure 127. Configuration Clock Source Selection

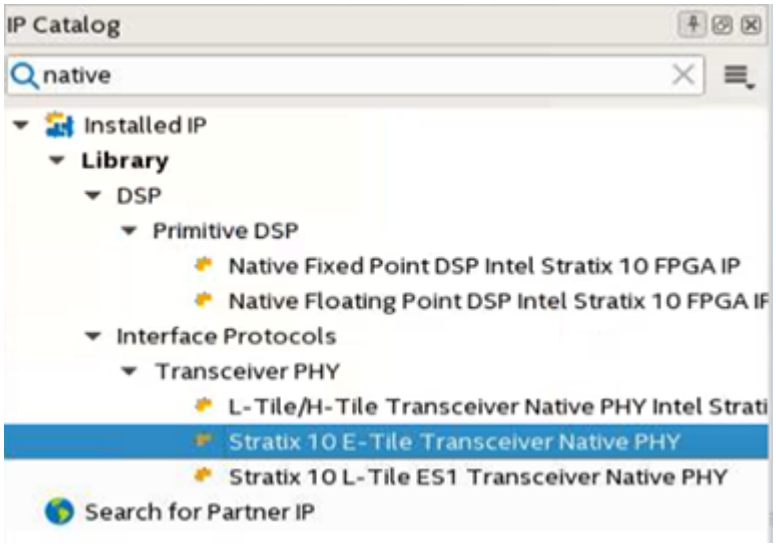


B.4. Instantiating the Transceiver Native PHY IP

This procedure describes how to instantiate your E-tile transceiver Native PHY IP core.

1. Locate the **E-Tile Transceiver Native PHY** IP core in the IP Catalog.

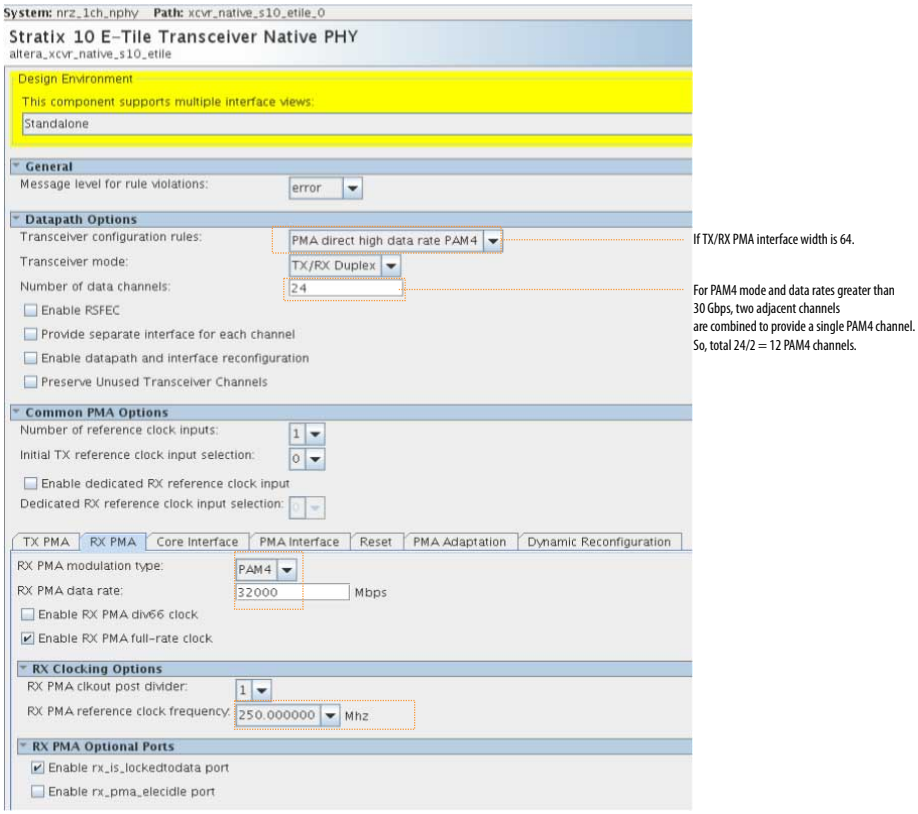
Figure 128. IP Catalog



The Native PHY IP Parameter Editor allows you to set many configurations, such as:

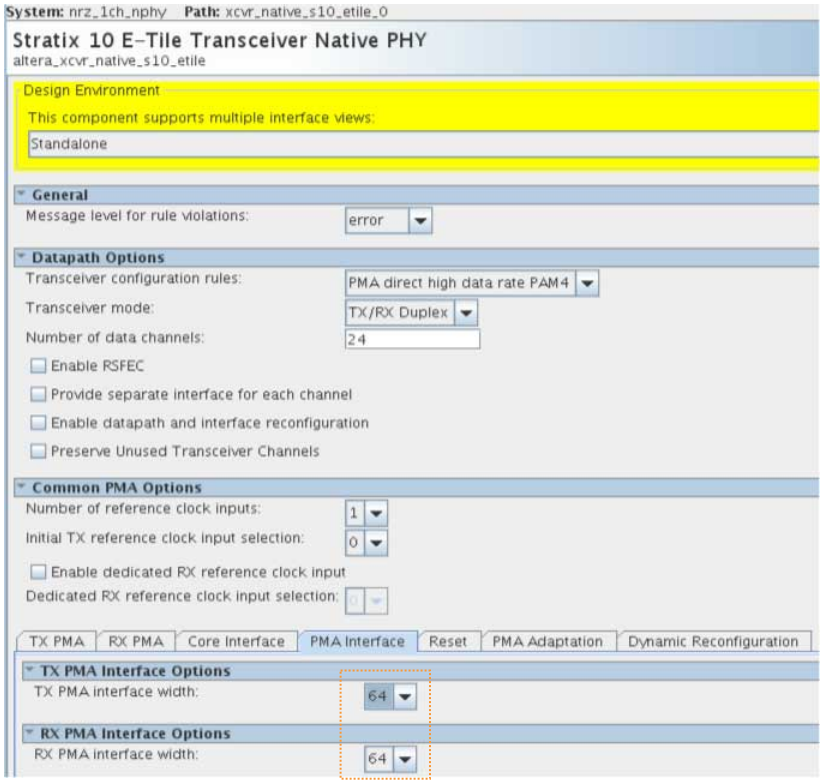
- Transceiver configuration rules (PMA direct or PMA direct high data rate PAM4)
- Number of data channels
- TX/RX PMA modulation type (NRZ or PAM4)
- TX/RX PMA data rate
- TX/RX PMA reference clock frequency

Figure 129. IP Parameter Editor Settings



Note: You must set the TX PMA tab as well.

Figure 130. PMA Interface Options for PMA Direct High Data Rate PAM4



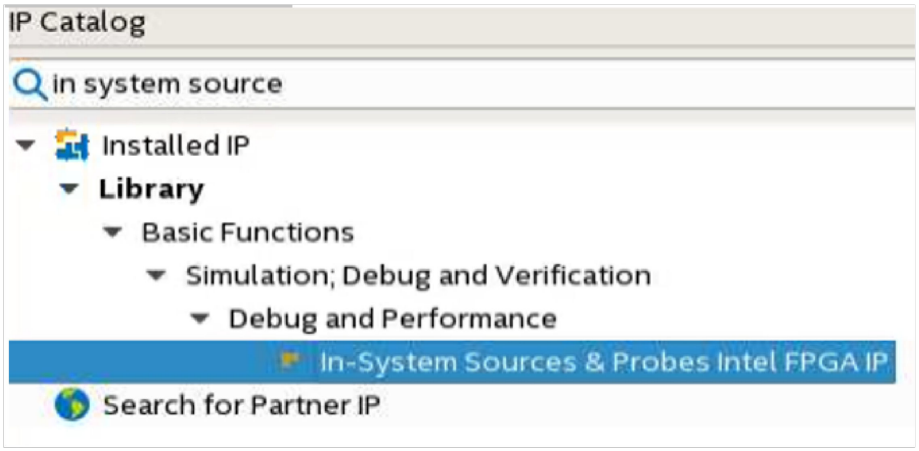
The PMA direct high data rate PAM4 transceiver configuration rule must select the 64 TX/RX PMA interface width.

B.5. Instantiating the In-system Sources and Probes Intel FPGA IP

This procedure describes how to instantiate the In-System Sources and Probes Intel FPGA IP core. This IP is used as a reset signal in *Making the Top Level Connection*.

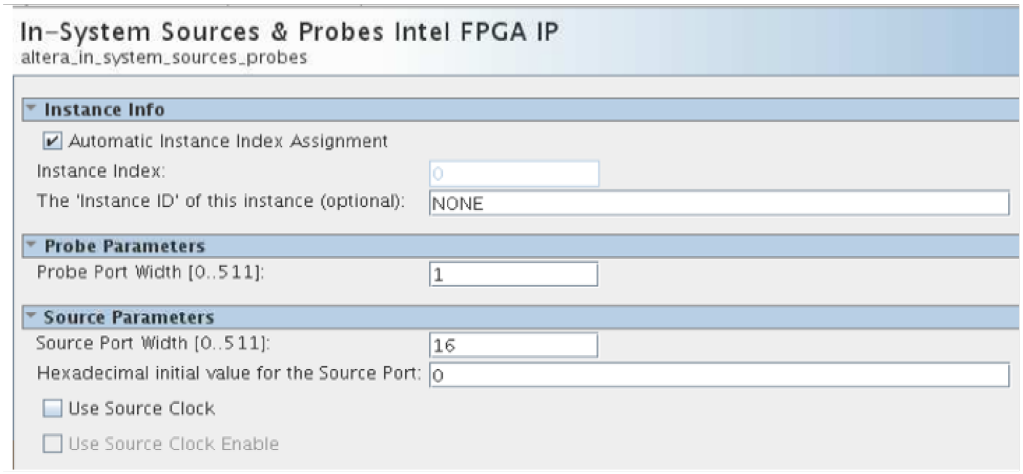
1. Type `In system source` in the IP Catalog search field.

Figure 131. IP Catalog Search Field



2. Double-click **In-System Sources & Probes Intel FPGA IP**.
3. Name the IP, `src`
4. Configure the IP with these settings.

Figure 132. In-System Source & Probes Intel FPGA IP Configuration



Related Information

Making the Top Level Connection on page 283

B.6. Making the Top Level Connection

Follow this procedure to make your top level connection through RTL.

1. Click **File > New** then select **Verilog HDL File**.
2. Write the RTL code to connect the blocks.

```
module pam4_12ch(
     input  wire        pll_refclk0,         //  pll_refclk0.clk
     input  wire        reset,
```

```
     input  wire [0:0]  reconfig_clk,        //   reconfig_clk.clk
     input  wire [23:0]  rx_serial_data,     //
rx_serial_data.rx_serial_data
     input  wire [23:0]  rx_serial_data_n,   //
rx_serial_data_n.rx_serial_data_n
     output wire [23:0]  tx_serial_data,     //
tx_serial_data.tx_serial_data
     output wire [23:0]  tx_serial_data_n    //
tx_serial_data_n.tx_serial_data_n
) ;

wire[15:0] source ;
assign reset = ~source[0] ;

wire [23:0]  rx_clkout, tx_clkout ;

nphy nphy (
     .rsfec_avmm2_avmmread_in(  ),     //    RSFEC_avmm2.read
     .rsfec_avmm2_avmmrequest_in(  ), //    .waitrequest
     .rsfec_avmm2_avmmwrite_in(  ),    //    .write
     .latency_sclk( 2'b0 ),            //    latency_sclk.latency_sclk
     .pll_refclk0( pll_refclk0 ),      //    pll_refclk0.clk
     .reconfig_write(  ),              //    reconfig_avmm.write
     .reconfig_read(  ),               //    .read
     .reconfig_address( ),             //    .address
     .reconfig_writedata( ),           //    .writedata
     .reconfig_readdata( ),            //    .readdata
     .reconfig_waitrequest( ),         //    .waitrequest
     .reconfig_clk( reconfig_clk ),    //     reconfig_clk.clk
     .reconfig_reset( ~reset ),        //     reconfig_reset.reset
     .reset( ~reset ),                 //     reset.reset
     .rx_clkout( rx_clkout ),          //     rx_clkout.clk
     .rx_coreclkin( rx_clkout ),       //     rx_coreclkin.clk
     .rx_dl_async_pulse(  ),           //
rx_dl_async_pulse.rx_dl_async_pulse
     .rx_dl_measure_sel(  ),           //
rx_dl_measure_sel.rx_dl_measure_sel
     .rx_is_lockedtodata(  ),          //
rx_is_lockedtodata.rx_is_lockedtodata
     .rx_parallel_data(  ),            //
rx_parallel_data.rx_parallel_data
     .rx_pma_ready(  ),                //     rx_pma_ready.rx_pma_ready
     .rx_ready(  ),                    //     rx_ready.rx_ready
     .rx_serial_data( rx_serial_data ),      //
rx_serial_data.rx_serial_data
     .rx_serial_data_n( rx_serial_data_n ),  //
rx_serial_data_n.rx_serial_data_n
     .tx_clkout( tx_clkout ),          //     tx_clkout.clk
     .tx_coreclkin( tx_clkout ),       //     tx_coreclkin.clk
     .tx_dl_async_pulse(  ),           //
tx_dl_async_pulse.tx_dl_async_pulse
     .tx_dl_measure_sel(  ),           //
tx_dl_measure_sel.tx_dl_measure_sel
     .tx_parallel_data( {12{48'b0,32'h0f0f0f0f, 48'b0, 32'h0f0f0f0f}} ), //
tx_parallel_data.tx_parallel_data
     .tx_pma_ready(  ),                //       tx_pma_ready.tx_pma_ready
     .tx_ready(  ),                    //       tx_ready.tx_ready
     .tx_serial_data( tx_serial_data ),        //
tx_serial_data.tx_serial_data
     .tx_serial_data_n( tx_serial_data_n )     //
tx_serial_data_n.tx_serial_data_n
    );

src src (
        .probe(  ),  //  probes.probe
        .source( source )  // sources.source
    );
```

```
endmodule
```

3. Verify the top level connections using the Netlist viewer.

B.7. Assigning Pins

Refer to the E-Tile Channel Placement tool to configure your transceiver channels.

Related Information

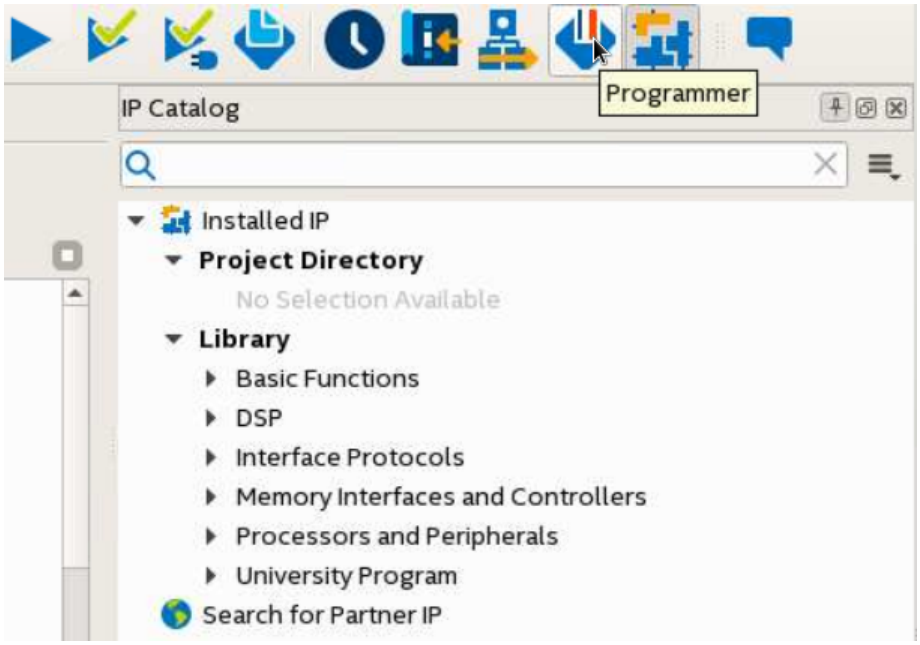
E-Tile Channel Placement Tool

B.8. Bringing up the Board

Follow this procedure to bring up your board.

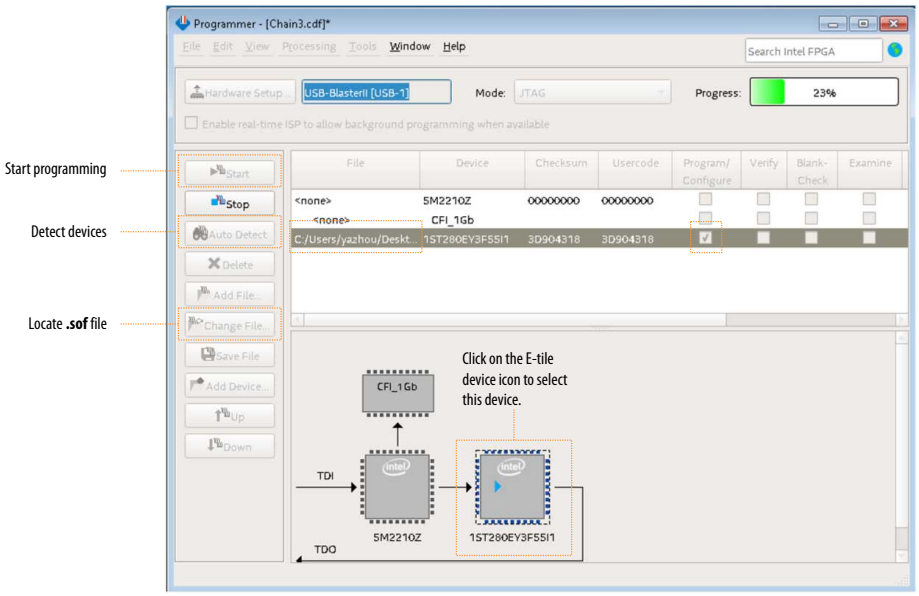
1. Set the refclk signal according to the guidelines provided in the *Device Family Pin Connection Guidelines*.
2. Download the settings to the board by clicking the **Programmer** tool.

Figure 133. Programmer Tool



3. Click **Auto Detect** to detect devices, then locate the your device.
4. Next to your device, click **Change File** to locate and add the **.sof** file.

Figure 134. Programmer Window



5. Select the **Program/Configure** option.
6. Click **Start** to begin programming.

Related Information

- Intel Stratix 10 Device Family Pin Connection Guidelines
- Intel Agilex Device Family Pin Connection Guidelines

B.9. Debug Tools

The Signal Tap Logic Analyzer helps you perform transceiver debug operations.

You can also use Transceiver Toolkit to perform transceiver debug operations. Refer to the *Intel Quartus Prime Pro Edition User Guide: Debug Tools* for more information about the Transceiver Toolkit.

Related Information

Intel Quartus Prime Pro Edition User Guide: Debug Tools

B.9.1. Monitoring Transceiver Signals

Signal Tap is a debug tool that allows you to monitor important transceiver-related signals, for example:

- `rx_is_lockedtodata`
- `rx_pma_ready`
- `rx_ready`
- `tx_pma_ready`
- `tx_ready`

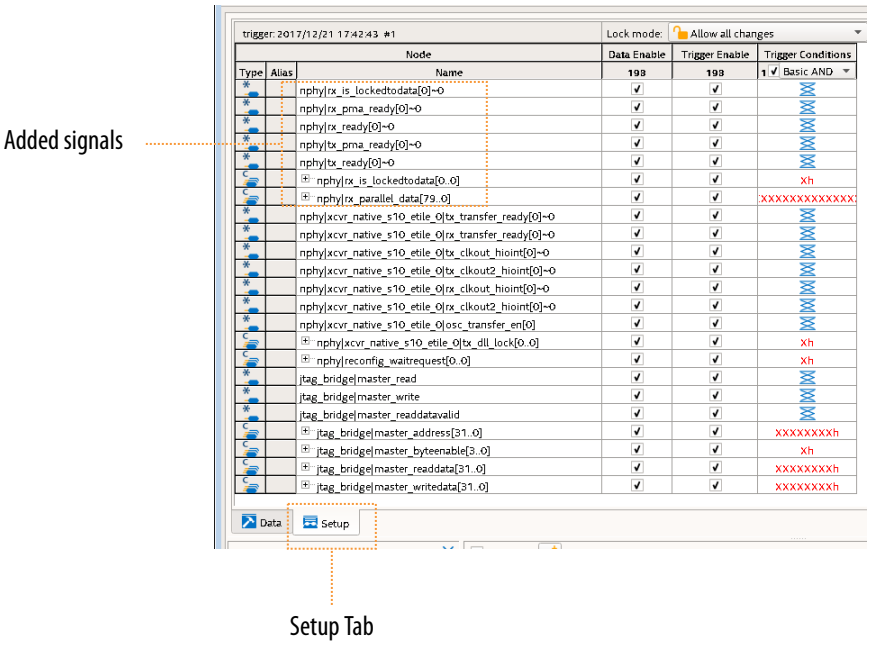
When all of these signals are high, the PMA is in the ready state and the receiver is locked to data.

To open Signal Tap, click **Tools > Signal Tap Logic Analyzer**.

Have your board powered on and running when you use Signal Tap. To add signals to probe:

1. Right-click anywhere in the **Setup** tab window and select **Add Nodes**.
2. Search for and insert your desired nodes.

Figure 135. Signal Tap Setup Tab



B.10. PMA Direct PAM4 30 Gbps to 57.8 Gbps Implementation Revision History

Document Version	Changes
2019.10.11	Made the following change: • Added the Related Information links for the Intel Agilex device documents.
2019.02.04	Made the following changes: • Updated GUI figures in *Instantiating the Transceiver Native PHY IP*.
2018.07.18	Made the following changes: • Added further description and a link to the Transceiver Toolkit documentation in the "Debug Tools" section.
2018.05.15	Initial release.

C. Signal Detect Algorithm

The physical channels to be supported are CEI-28 VSR, a CAUI-4 chip-to-module connection and CEI-56 VSR (PAM4). Other physical channels can be used depending on the application.

The receiver channel must autonomously adapt and lock to the incoming signal when a valid signal becomes present. "Autonomously" means without assistance or information from downstream processing or upper protocol layers, in other words, based exclusively on information available to and via the transceiver itself.

An invalid signal is any situation where the incoming signal is not valid, typically:

- Module unplugged
- Fiber unplugged on module
- No signal received on fiber
- Bad signal received on fiber

The receiver adaptation completes without disrupting the transmitter path.

The adaptation results in an acceptable BER, for example:

- CEI-28 VSR: BER $1x10^{-15}$
- CEI-56 VSR: BER $1x10^{-6}$

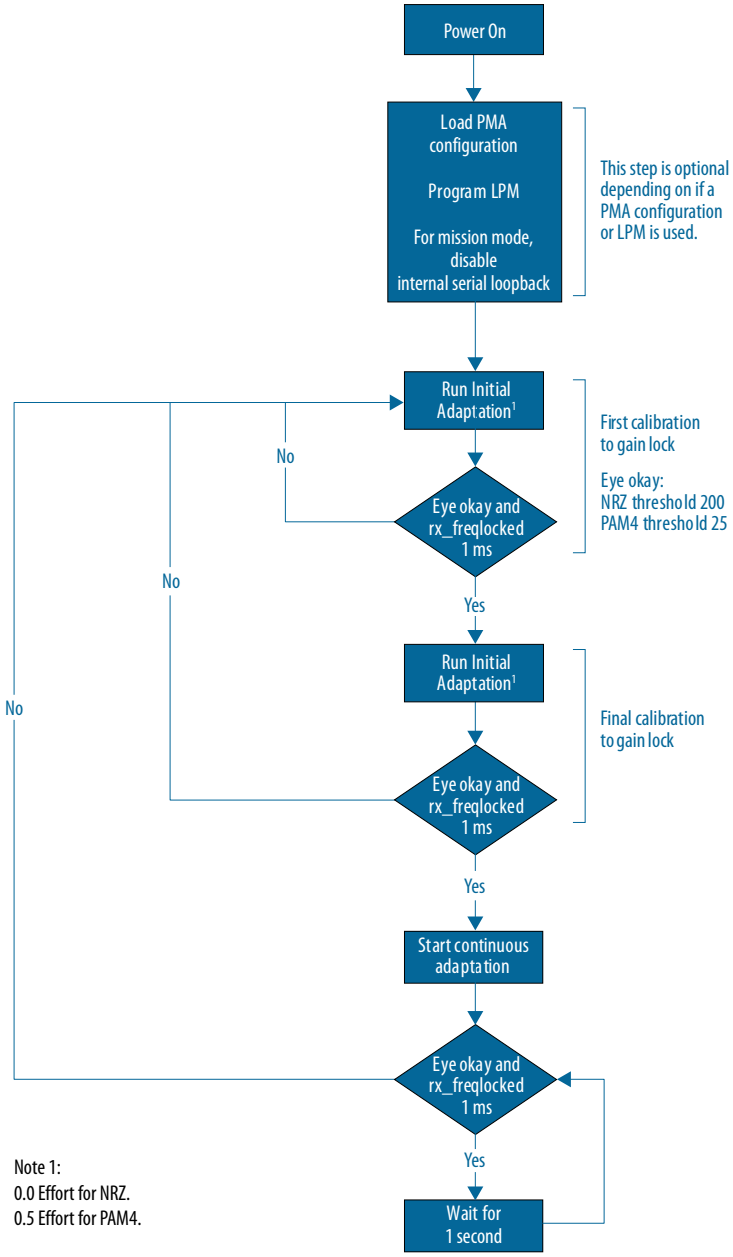
The adaptation completes in:

- Approximately 100 ms for NRZ
- Approximately 4 s for PAM4

The entire adaptation process must be able to operate in low power mode. The signal quality is maintained provided that the die temperature variation does not exceed 3 °C per minute and continuous adaptation is enabled.

This process does not deadlock to an unrecoverable state.

Figure 136. Signal Detect Algorithm Flowchart



Valid Signal

For a signal is valid, both of these conditions must be met:

- `rx_freqlocked_1ms`: A filtered version of `rx_is_lockedtodata`. "Filtered" means that it deasserts instantly on loss of lock and `rx_is_lockedtodata` be asserted continuously without any deassertions for at least 1 ms before `rx_freqlocked_1ms` asserts.
- Eye okay: Requires a read from the transceiver via AVMM and compares a value with an acceptable threshold value of 200 (NRZ) or 25 (PAM4). These units are expressed in steps. The eye threshold values are application-dependent and should be determined by your system validation. See *Reading NRZ/PAM4 Eye Height*.

If at any point in the algorithm the above valid condition is not met, the adaptation complete status becomes false, and the flow returns to the initial adaptation stage.

The loss of a valid signal also results in an interrupt notification to the downstream datapath and at the system level.

Initial Adaptation Stage

The assumption at the start of the test flow is that there may be no signal or an invalid signal may be present. Until an initial adaptation has been run, there is no way to reliably achieve lock, so the initial adaptation process is repeatedly triggered while waiting for a valid signal as per the above definition.

It is possible that the first time a valid signal is detected, the adaptation may not be optimal, for example, because the signal appeared during the initial adaptation process. Therefore, an additional initial adaptation is run following this first detection.

If the signal is still valid after the second initial adaptation, that part of the test flow is complete. When completed, a notification (signal or interrupt) is provided at the system level and to the downstream datapath.

The operations in this process take:

- Reading eye height for an eye okay check: <50 µs
- Initiating initial adaptation: typically, 680 µs, sometimes 3 ms

As the completion time for the initial adaptation varies, the algorithm runs the loop on a fixed timing, triggering a new initial adaptation every 40 ms (chosen to provide some margin) if the loop exit conditions have not been met.

Rarely, if many transceiver channels are in the initial adaptation loop and many of these take a long time to respond on the bus, there may not be enough time to service all channels within this time. In this case, the loop takes longer to complete then continues as usual.

This gives an average completion time of 100 ms, typically, varying between 80 and 120 ms.

For PAM4, the observed completion time is approximately 1.5 seconds using an effort of 0.5.

Ongoing Adaptation Stage

In order to maintain good receiver equalization through temperature variations (within specification), an ongoing, non-traffic affecting adaptation process (continuous adaptation) is run while the signal is valid.

Each continuous adaptation takes approximately one second to complete and is repeated automatically while in a continuous mode.

If the signal is removed while in this mode and multiple continuous adaptation operations are performed in this state, the receiver equalization may gradually enter a bad state which cannot be recovered from by re-running an initial adaptation and reaches a deadlock from which this algorithm cannot recover.

To prevent this, the algorithm monitors the `rx_freqlocked_1ms` signal and immediately reverts to the initial adaptation stage, which issues an initial adaptation which effectively stops the continuous adaptation. Additionally, the eye okay status is polled on every second, and, if no longer meeting the criterion, reverts to the initial adaptation stage.

Therefore, at most, one or two continuous adaptation operations execute on an invalid signal which is not enough to bring the transceiver into a bad state.

Internal Serial Loopback

Internal serial loopback requires a different adaptation and cannot be used in LPM. However, there needs to be the ability to switch between internal serial loopback and non-internal serial loopback and perform appropriate adaptation each time.

Internal serial loopback requires *Resetting the RX Equalization* after setting the internal serial loopback. This RX equalization reset does not impact the transmitter.

The adaptation algorithm must be exited before configuring internal serial loopback and restarted after disabling internal serial loopback and enabling LPM.

Related Information

- Reading NRZ/PAM4 Eye Height on page 233
- Resetting the RX Equalization on page 209
- 2x Four-Channel Dual Mode 28.3 Gbps Soft PRBS Test Design with Adaptation Soft IP
- 2x Eight-Channel PAM4 58 Gbps Soft PRBS Test Design with Adaptation Soft IP

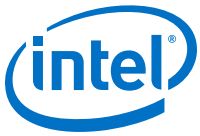
C.1. Signal Detect Algorithm Revision History

Document Version	Changes
2020.01.31	Made the following change: • Updated the "Signal Detect Algorithm Flowchart." • Added links to example designs.
2019.10.11	Initial release.

D. Detailed Steps for Reconfiguring from Mission Mode to Channel Protection Mode

These steps target the channel that is to be set into channel protection mode. If using PAM4 for mission mode, turn on **Preserve Unused Transceiver Channels**.

1. Assert `tx_reset` or `rx_reset`.
2. Wait for the `tx_ready` or `rx_ready` to deassert.
3. Disable the PMA with PMA attribute code 0x0001.
   a. Write 0x8A[7] = 0x1 to ensure that the PMA attribute status flag (0x8A[7] for the previous attribute) is cleared before writing to registers 0x84 through 0x87 to load in the new PMA attribute.
   b. Write 0x84[7:0] = 0x00.
   c. Write 0x85[7:0] = 0x00.
   d. Write 0x86[7:0] = 0x01.
   e. Write 0x87[7:0] = 0x00.
   f. Write 0x90[0] = 1'b1.
   g. Read 0x8A[7]. It should be 1.
   h. Read 0x8B[0] until it changes to 0.
   i. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
   j. Wait for `tx_pma_ready` or `rx_pma_ready` to deassert.
4. Reset the internal controller inside the PMA.
   a. Write 0x200[7:0] = 0x00.
   b. Write 0x201[7:0] = 0x00.
   c. Write 0x202[7:0] = 0x00.
   d. Write 0x203[7:0] = 0x81.
   e. Read 0x207 until it becomes 0x80. This indicates that the operation completed successfully.
5. Set the TX and RX encoding to NRZ mode and the TX and RX width to 32 bits.
   a. Write 0x84[7:0] = 0x55.
   b. Write 0x85[7:0] = 0x00.
   c. Write 0x86[7:0] = 0x14.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.



h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.

6. Set the TX baud rate.
   a. Compute the desired TX reference clock multiplier as the closest valid value to `2500/refclock_freq_in_MHz`. For example, if the reference clock frequency is 156.25 MHz, the computed reference clock multiplier is 16 which is a valid multiplier. Use the same value for the TX and RX reference clock multipliers.
   b. Write 0x84[7:0] = 0x10 for the decimal 16 in this example.
   c. Write 0x85[7:0] = 0x10.
   d. Write 0x86[7:0] = 0x05.
   e. Write 0x87[7:0] = 0x00.
   f. Write 0x90[0] = 1'b1.
   g. Read 0x8A[7]. It should be 1.
   h. Read 0x8B[0] until it changes to 0.
   i. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
7. Set the RX baud rate.
   a. Compute the desired TX reference clock multiplier as the closest valid value to `2500/refclock_freq_in_MHz`. For example, if the reference clock frequency is 156.25 MHz, the computed reference clock multiplier is 16, which is a valid multiplier. Use the same value for the TX and RX reference clock multipliers.
   b. Write 0x84[7:0] = 0x10 for the decimal 16 in this example.
   c. Write 0x85[7:0] = 0x10.
   d. Write 0x86[7:0] = 0x06.
   e. Write 0x87[7:0] = 0x00.
   f. Write 0x90[0] = 1'b1.
   g. Read 0x8A[7]. It should be 1.
   h. Read 0x8B[0] until it changes to 0.
   i. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
8. Set the PRBS control to PRBS7.
   a. Write 0x84[7:0] = 0x20.
   b. Write 0x85[7:0] = 0x01.
   c. Write 0x86[7:0] = 0x02.
   d. Write 0x87[7:0] = 0x00.
   e. Write 0x90[0] = 1'b1.
   f. Read 0x8A[7]. It should be 1.
   g. Read 0x8B[0] until it changes to 0.
   h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
9. Set the PMA to internal serial loopback.
   a. Write 0x84[7:0] = 0x01.
   b. Write 0x85[7:0] = 0x01.
   c. Write 0x86[7:0] = 0x08.

d. Write 0x87[7:0] = 0x00.
e. Write 0x90[0] = 1'b1.
f. Read 0x8A[7]. It should be 1.
g. Read 0x8B[0] until it changes to 0.
h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.

10. Before entering channel protection mode, save the mission mode settings for the mission mode channels with values from addresses 0x05, 0x07, and 0x38.
11. Set up the transceiver interface DCC. Use read-modify-write to change only bits 1:0.
    a. Write 0x38[1:0] = 2'b01.
12. Set up the RX clocks. Use read-modify-write to set only bits 7 and 1.
    a. Write 0x07[7]=1'b0.
    b. Write 0x07[1]=1'b0.
13. Set up the TX clocks. Use read-modify-write to change only bits 7:2.
    a. Write 0x05[7:2]=6'b000010.
14. Enable the TX and RX of the PMA with PMA attribute code 0x0001. This disables the TX driver for this channel.
    a. Write 0x84[7:0] = 0x03.
    b. Write 0x85[7:0] = 0x00.
    c. Write 0x86[7:0] = 0x01.
    d. Write 0x87[7:0] = 0x00.
    e. Write 0x90[0] = 1'b1.
    f. Read 0x8A[7]. It should be 1.
    g. Read 0x8B[0] until it changes to 0.
    h. Write 0x8A[7] to 1'b1 to clear the 0x8A[7] value.
    i. Wait for `tx_pma_ready` or `rx_pma_ready` to assert.
15. Run initial adaptation.

D.1. Detailed Steps for Reconfiguring from Mission Mode to Channel Protection Mode Revision History

Document Version	Changes
2020.01.31	Initial release.