



Networking Interface for Open Programmable Acceleration Engine

Intel® Programmable Acceleration Card with Intel® Arria® 10 GX FPGA

Updated for Intel® Acceleration Stack for Intel® Xeon® CPU with FPGAs: **1.2**



UG-20188 | 2019.08.05

Latest document on the web: [PDF](#) | [HTML](#)



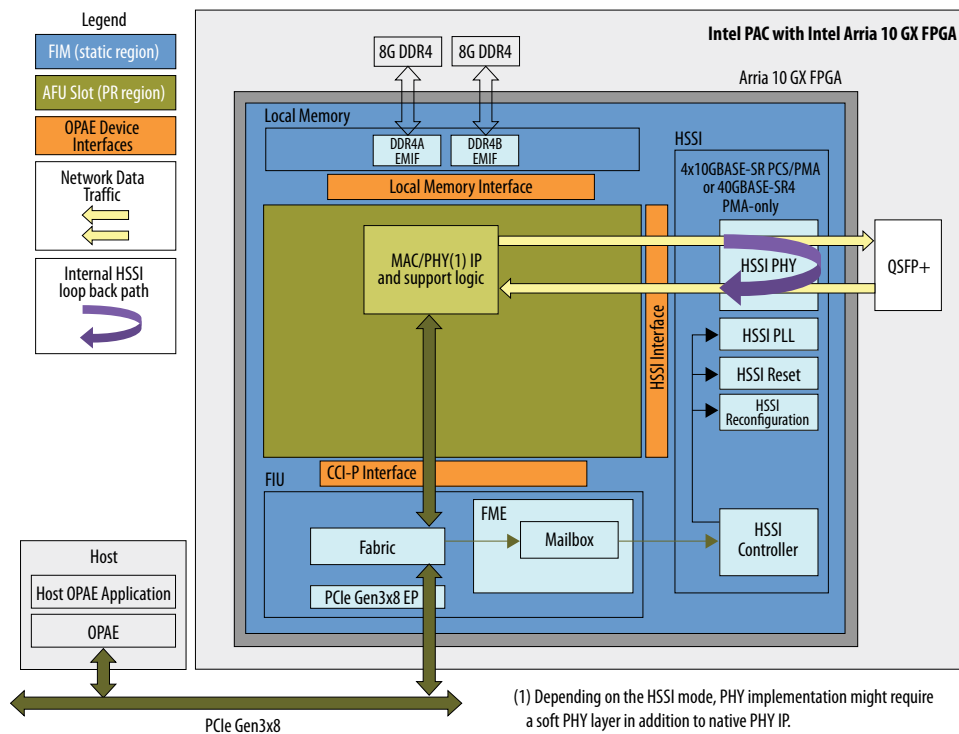
Contents

1. Introduction.....	3
1.1. How to Use this Guide.....	4
2. AFU Design.....	5
2.1. HSSI Device Interface.....	5
2.1.1. HSSI Clocks.....	6
2.1.2. HSSI Unified Data Interface.....	7
2.1.3. HSSI PHY Control and Status.....	8
2.1.4. HSSI PR Management.....	9
2.1.5. HSSI Reset Control and Status.....	10
2.1.6. Initialization.....	10
2.2. Connecting the MAC to the HSSI PHY.....	11
2.2.1. 4x10GBASE-SR Mode.....	11
2.2.2. 40GBASE-SR4 Mode.....	14
2.3. Verifying Network Port Function.....	17
3. OPAE Support.....	18
3.1. OPAE Driver HSSI sysfs Files.....	18
3.1.1. The board_id sysfs.....	18
3.1.2. The eeprom sysfs.....	18
3.1.3. The config sysfs.....	19
3.1.4. The equalizer_tune sysfs.....	19
3.2. Managing the Network Port.....	24
3.2.1. Configuring the Network Port.....	24
3.2.2. Reading the Base MAC Address from the Intel PAC with Intel Arria 10 GX FPGA.....	25
3.2.3. Modifying HSSI PHY Transceiver PMA Settings.....	25
4. Networking Interface for Open Programmable Acceleration Engine: Intel PAC with Intel Arria 10 GX FPGA Revision History.....	27

1. Introduction

The Intel® Programmable Acceleration Card with Intel Arria® 10 GX FPGA features a QSFP+ network port that can be configured for either 4x10GBASE-SR or 40GBASE-SR4 operation. This guide documents how to design for the network port feature in an accelerator functional unit (AFU) design and how to provision it from the host using the Open Programmable Acceleration Engine (OPAE) driver and tools. The following figure overviews the Intel PAC with Intel Arria 10 GX FPGA OPAE hardware platform.

Figure 1. Overview of the Intel PAC with Intel Arria 10 GX FPGA



Host/client-side network packet data passes through the Core Cache Interface (CCI-P) to MAC/PHY IP implemented in the AFU, which interfaces to the high speed serial interface (HSSI) PHY in the FPGA interface manager (FIM) through the `hssi` device interface. The host configures the HSSI PHY and retrieves MAC address information using the OPAE kernel driver. The OPAE kernel driver communicates with the HSSI Controller in the FIM through the FPGA management engine (FME) mailbox, to initiate configuration and requests for information (including the MAC address).

For detailed information about the FME, refer to the *Intel Acceleration Stack Quick Start Guide for Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA*.



Related Information

[Intel Acceleration Stack Quick Start Guide for Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA](#)

1.1. How to Use this Guide

There are two major components to enable using the network port feature on the Intel PAC with Intel Arria 10 GX FPGA. First, the host must configure the HSSI PHY in the hardware platform's FIM for one of the supported modes (4x10GBASE-SR or 40GBASE-SR4). Secondly, the host must load an accelerator function (AF) that supports the network port feature.

The *AFU Design* section covers the requirements in the AFU design to enable the network port feature. The section describes the network MAC and PHY components that must be implemented in the AFU design and how to connect them to the HSSI PHY in the FIM through the `hssi` device interface. This section also gives some guidance on verifying your AFU implementation for network port connectivity in hardware.

The *OPAE Support* section covers using the OPAE driver and tools to provision the network port feature from the host, including configuring the HSSI PHY for the desired mode of operation, loading a network-enabled AF, and retrieving information from the Intel PAC with Intel Arria 10 GX FPGA such as MAC address.

This document was previously titled as *HSSI User Guide for Intel Programmable Acceleration Card (PAC) Intel Arria 10 GX FPGA*.

Related Information

- [AFU Design](#) on page 5
- [OPAE Support](#) on page 18

2. AFU Design

To enable the network port feature on the Intel PAC with Intel Arria 10 GX FPGA, the AFU must implement the MAC and PHY IP blocks shown in the following table:

Table 1. Required MAC and PHY IP Blocks

Network Port Mode	Required IP Blocks in the AFU	HSSI PHY Mode
4x10GBASE-SR	One 10GbE MAC per channel supported by the AFU (up to 4)	4x10GBASE-SR PCS/PMA
40GBASE-SR4	One 40GbE MAC One 40GbE Physical Coding Sublayer (PCS) PHY	40GBASE-SR4 PMA-only

In addition to the above network IP blocks, the AFU must also implement the following supporting infrastructure:

- Client-side data interfaces and DMA required to move data between host or local memory, AFU workload streams, and the network port
- MMIO access through the `cci-p` device interface for host access to MAC/PHY control and status registers (CSRs), network statistics, and similar information

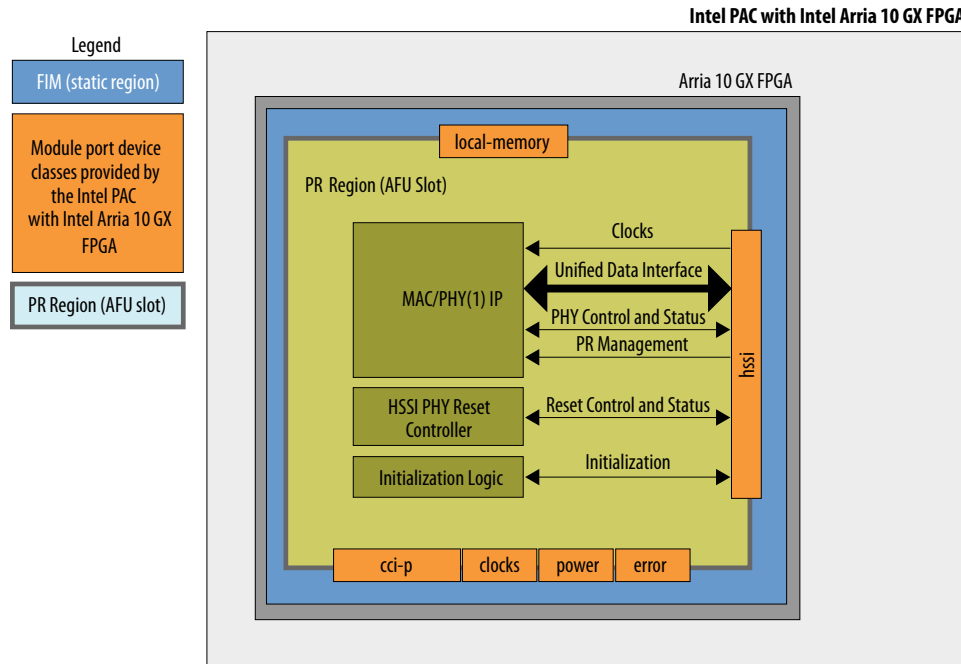
The FIM provides clock resources for client and PHY interfaces through the `cci-p` and `hssi` device interfaces.

The remainder of this section describes the `hssi` device interface and how to connect MAC and PHY IP implemented in the AFU to the HSSI PHY using the `hssi` interface.

2.1. HSSI Device Interface

AFUs interface with the network port on the Intel PAC with Intel Arria 10 GX FPGA using the `hssi:raw_pr` device interface, which is shown in the below high level interface block diagram.

Figure 2. Overview of MAC/PHY IP Connection to hssi Interface



(1) Depending on the HSSI mode, PHY implementation might require a soft PHY layer in addition to native PHY IP.

A unified data interface connects the network port to the MAC/PHY IP. This can be Intel FPGA IP, third-party IP, or your own proprietary IP. The unified data interface consists of a fixed set of physical ports that are mapped to specific signaling functions based on the configured HSSI PHY mode. The `hssi:raw_pr` interface also provides clocks for synchronization and signaling support for HSSI PHY management.

The AFU must implement reset logic for the HSSI PHY using the reset control and status signaling provided by the `hssi:raw_pr` interface. See the *HSSI Reset Control and Status* section for more information.

The `pr_hssi_if.vh` SystemVerilog* header defines the `hssi:raw_pr` interface and is located in the Intel PAC with Intel Arria 10 GX FPGA hardware platform database library within the OPAAE SDK:

```
$OPAAE_PLATFORM_ROOT/hw/lib/build/platform/pr_hssi_if.vh
```

The sections that follow detail the ports included in the `hssi:raw_pr` interface.

Related Information

[HSSI Reset Control and Status](#) on page 10

2.1.1. HSSI Clocks

The clocks of the `hssi` interface synchronize the unified data interface between the MAC/PHY IP and the HSSI PHY.



Table 2. HSSI Clocks

Signal directions listed for hssi ports are from the perspective of the FIM.

hssi Port Name	Width	Direction	4x10GBASE-R Mode Description	40GBASE-SR4 Mode Description
f2a_tx_clk	1	Output	A 156.25MHz clock derived from the HSSI PHY's clock generation block (CGB) tx_pma_div_clkout clock output. All transmit data and control from the MAC to the HSSI PHY is synchronous to f2a_tx_clk.	A 312.5MHz clock derived from the HSSI PHY's CGB tx_pma_div_clkout clock output. All transmit data and control from the MAC/PHY to the HSSI PHY is synchronous to f2a_tx_clk.
f2a_tx_clkx2	1	Output	A 312.5MHz clock derived from the HSSI PHY's CGB tx_pma_div_clkout clock output and phase-aligned with f2a_tx_clk.	A 312.5MHz clock derived from the PHY's CGB tx_pma_div_clkout clock output and phase-aligned with f2a_tx_clk.
f2a_tx_locked	1	Output	Locked status for f2a_tx_clk and f2a_tx_clkx2.	Locked status for f2a_tx_clk and f2a_tx_clkx2.
f2a_rx_clk_ln0	1	Output	A 156.25MHz clock derived from the HSSI PHY's transmitter and receive CDR PLL clock input reference. All receive data and control from the HSSI PHY to the MAC is synchronous to f2a_rx_clk_ln0.	A 312.5MHz clock derived from the HSSI PHY's receive CDR in lane 0. All receive data and control from the HSSI PHY to the MAC/PHY is synchronous to f2a_rx_clk_ln0.
f2a_rx_clkx2_ln0	1	Output	A 312.5MHz clock derived from the HSSI PHY's transmitter and receive CDR PLL clock input reference and phase-aligned with f2a_rx_clk_ln0.	A 312.5MHz clock derived from the HSSI PHY's receive CDR in lane 0 and phase-aligned with f2a_rx_clk_ln0.
f2a_rx_locked_ln0	1	Output	Locked status for f2a_rx_clk_ln0 and f2a_rx_clkx2_ln0.	Locked status for f2a_rx_clk_ln0 and f2a_rx_clkx2_ln0.
f2a_rx_clk_ln4	1	Output	Reserved	Reserved
f2a_rx_locked_ln4	1	Output	Reserved	Reserved

2.1.2. HSSI Unified Data Interface

The HSSI unified data interface conforms to the Arria 10 FPGA Transceiver Native PHY IP with enhanced PCS. It consists of generic parallel data and encoding control interfaces for transmit and receive that are mapped to specific signaling behavior based on the configured HSSI PHY mode. The unified data interface also includes flow control ports to manage passing data to and from the HSSI PHY.

The below table provides a cross reference from the hssi:raw_pr unified data interface signals to the Arria 10 FPGA Transceiver Native PHY IP with enhanced PCS signal set. For detailed information on these signals, see the [Intel Arria 10 Transceiver PHY User Guide](#) as referenced in the below table.

Table 3. HSSI Unified Data Interface

hssi Port Name	Width	Direction	Clock Domain	Native PHY IP Port Name	Reference in Intel Arria 10 Transceiver PHY User Guide
Transmit and Receive Data and Encoding Control Ports					
a2f_tx_parallel_data	(4*128)	Input	f2a_tx_clk	tx_parallel_data	Table <i>Enhanced TX PCS: Parallel Data, Control, and Clocks</i> in Enhanced PCS Ports
a2f_tx_control	(4*18)	Input	f2a_tx_clk	tx_control	
f2a_rx_parallel_data	(4*128)	Output	f2a_rx_clk_l1n0	rx_parallel_data	
f2a_rx_control	(4*20)	Output	f2a_rx_clk_l1n0	rx_control	
Flow Control Ports					
f2a_tx_enh_fifo_full	4	Output	f2a_tx_clk	tx_enh_fifo_full	Table <i>Enhanced PCS TX FIFO</i> in Enhanced PCS Ports
f2a_tx_enh_fifo_pfull	4	Output	f2a_tx_clk	tx_enh_fifo_pfull	
f2a_tx_enh_fifo_empty	4	Output	f2a_tx_clk	tx_enh_fifo_empty	
f2a_tx_enh_fifo_empty	4	Output	f2a_tx_clk	tx_enh_fifo_empty	
a2f_tx_enh_data_valid	4	Input	f2a_tx_clk	tx_enh_data_valid	
f2a_rx_enh_fifo_full	4	Output	f2a_rx_clk_l1n0	rx_enh_fifo_full	Table <i>Enhanced PCS RX FIFO</i> in Enhanced PCS Ports
f2a_rx_enh_fifo_pfull	4	Output	f2a_rx_clk_l1n0	rx_enh_fifo_pfull	
f2a_rx_enh_fifo_empty	4	Output	f2a_rx_clk_l1n0	rx_enh_fifo_empty	
f2a_rx_enh_fifo_empty	4	Output	f2a_rx_clk_l1n0	rx_enh_fifo_empty	
f2a_rx_enh_data_valid	4	Output	f2a_rx_clk_l1n0	rx_enh_data_valid	
a2f_rx_enh_fifo_ready	4	Input	f2a_rx_clk_l1n0	rx_enh_fifo_ready	

Related Information

[Intel Arria 10 Transceiver PHY User Guide](#)

2.1.3. HSSI PHY Control and Status

This set of ports on the hssi interface provide for HSSI PHY receive Physical Medium Attachment (PMA) clock data recovery (CDR) lock sequencing control, PCS status, and transceiver loopback control. The signaling behavior conforms to the Arria 10 FPGA Transceiver Native PHY IP with enhanced PCS. The below table cross references the hssi port names to the Native PHY IP port names.



Table 4. HSSI PHY Control and Status Ports

For detailed information on these signals, see the *Intel Arria 10 Transceiver PHY User Guide* as referenced in the below table.

hssi Port Name	Width	Direction	Clock Domain	Native PHY IP Port Name	Reference in <i>Intel Arria 10 Transceiver PHY User Guide</i>
a2f_rx_serialpbken	4	Input	Async	rx_serialpbken	Table TX PMA Ports in PMA Ports
a2f_rx_set_locktoref	4	Input	Async	rx_set_locktoref	<ul style="list-style-type: none"> Table RX PMA Ports in PMA Ports Resetting Transceiver Channels
f2a_rx_is_lockedtoref	4	Output	Async	rx_is_lockedtoref	
a2f_rx_set_locktodata	4	Input	Async	rx_set_locktodata	
f2a_rx_enh_blk_lock	4	Output	f2a_rx_clk_ln0	rx_enh_blk_lock	Table Block Synchronizer in Enhanced PCS Ports
f2a_rx_enh_highber	4	Output	f2a_rx_clk_ln0	rx_enh_highber	Table 10GBASE-R BER Checker in Enhanced PCS Ports

Related Information

[Intel Arria 10 Transceiver PHY User Guide](#)

2.1.4. HSSI PR Management

The f2a_prmgmt_ctrl_clk port is a 100MHz free running clock source. The MAC/PHY IP and related AFU logic can use this clock for lower speed logic. The f2a_prmgmt_ram_ena port is used as a reset source from the HSSI PHY to PCS PHY IP in the AFU.

The remaining ports on the PR management bus are for internal use in Intel AFU example designs.

Table 5. HSSI PR Management Port Characteristics

hssi Port Name	Width	Direction	Clock Domain	Description
f2a_prmgmt_ctrl_clk	1	Output		Optional low-speed clock source
a2f_prmgmt_fatal_err	1	Input	f2a_prmgmt_ctrl_clk	Leave outputs disconnected and drive inputs low.
a2f_prmgmt_dout	32	Input	f2a_prmgmt_ctrl_clk	
f2a_prmgmt_cmd	16	Output	f2a_prmgmt_ctrl_clk	
f2a_prmgmt_addr	16	Output	f2a_prmgmt_ctrl_clk	
f2a_prmgmt_din	32	Output	f2a_prmgmt_ctrl_clk	
f2a_prmgmt_freeze	1	Output	f2a_prmgmt_ctrl_clk	
f2a_prmgmt_arst	1	Output	Async	
f2a_prmgmt_ram_ena	1	Output	Async	

2.1.5. HSSI Reset Control and Status

The Reset Control and Status ports conform to the Native PHY IP with enhanced PCS defined signal behavior. The below table cross references the `hssi` port names to the Native PHY IP port names.

Table 6. HSSI Reset Control and Status Port Characteristics

For detailed information on these signals, see the *Intel Arria 10 Transceiver PHY User Guide* as referenced in the below table.

hssi Port Name	Width	Direction	Clock Domain	Native PHY IP Port Name	Reference in <i>Intel Arria 10 Transceiver PHY User Guide</i>
a2f_tx_analogreset	4	Input	Async	tx_analogreset	<ul style="list-style-type: none"> Table <i>Reset Ports</i> in <i>PMA Ports</i> Resetting Transceiver Channels
a2f_tx_digitalreset	4	Input	Async	tx_digitalreset	
a2f_rx_analogreset	4	Input	Async	rx_analogreset	
a2f_rx_digitalreset	4	Input	Async	rx_digitalreset	
f2a_tx_cal_busy	1	Output	Async	tx_cal_busy	<ul style="list-style-type: none"> Table <i>Calibration Status Ports</i> <i>PMA Ports</i> Resetting Transceiver Channels
f2a_tx_pll_locked	1	Output	Async	pll_locked	
f2a_rx_cal_busy	1	Output	Async	rx_cal_busy	
f2a_rx_is_lockedtoday	4	Output	Async	rx_is_lockedtoday	<ul style="list-style-type: none"> Table <i>RX PMA Ports</i> in <i>PMA Ports</i> Resetting Transceiver Channels

Related Information

[Intel Arria 10 Transceiver PHY User Guide](#)

2.1.6. Initialization

The initialization handshake controls can optionally be used to sequence readiness between the MAC/PHY IP in the AFU and HSSI PHY mode completion.

Table 7. Initialization Handshake Control Ports

hssi Port Name	Width	Direction	Clock Domain	Description
a2f_init_start	1	Input	Async	Signal to indicate AFU ready (optional)
f2a_init_done	1	Output	Async	Signal to indicate HSSI PHY initialization to chosen mode complete (optional)



2.2. Connecting the MAC to the HSSI PHY

The OPAE SDK includes the following two sample AFUs that show how to connect MAC and PHY IP to the `hssi` interface:

Table 8. Sample AFUs and Documentation

Network Port Mode	Documentation	Sample AFU Location
4x10GBASE-SR	10Gbps Ethernet Accelerator Functional Unit (AFU) Design Example User Guide	<code>\$OPAE_PLATFORM_ROOT/hw/samples/eth_e2e_e10</code>
40GBASE-SR4	40Gbps Ethernet Accelerator Functional Unit (AFU) Design Example User Guide	<code>\$OPAE_PLATFORM_ROOT/hw/samples/eth_e2e_e40</code>

Use the information in this section together with the sample AFUs for guidance on connecting MAC and PHY IP to the `hssi` interface.

The sections that follow contain connection diagrams that utilize pseudo RTL code to define connectivity on buses using Verilog*-2001 indexed part selects.

For example:

```
logic port_name[15:0]; //All hssi port vectors are little endian.  
assign port_name[8 +:8] = {8{1'b0}};
```

assigns all zeros to the upper eight bits ([15:8]) of the 16-bit vector, `port_name`.

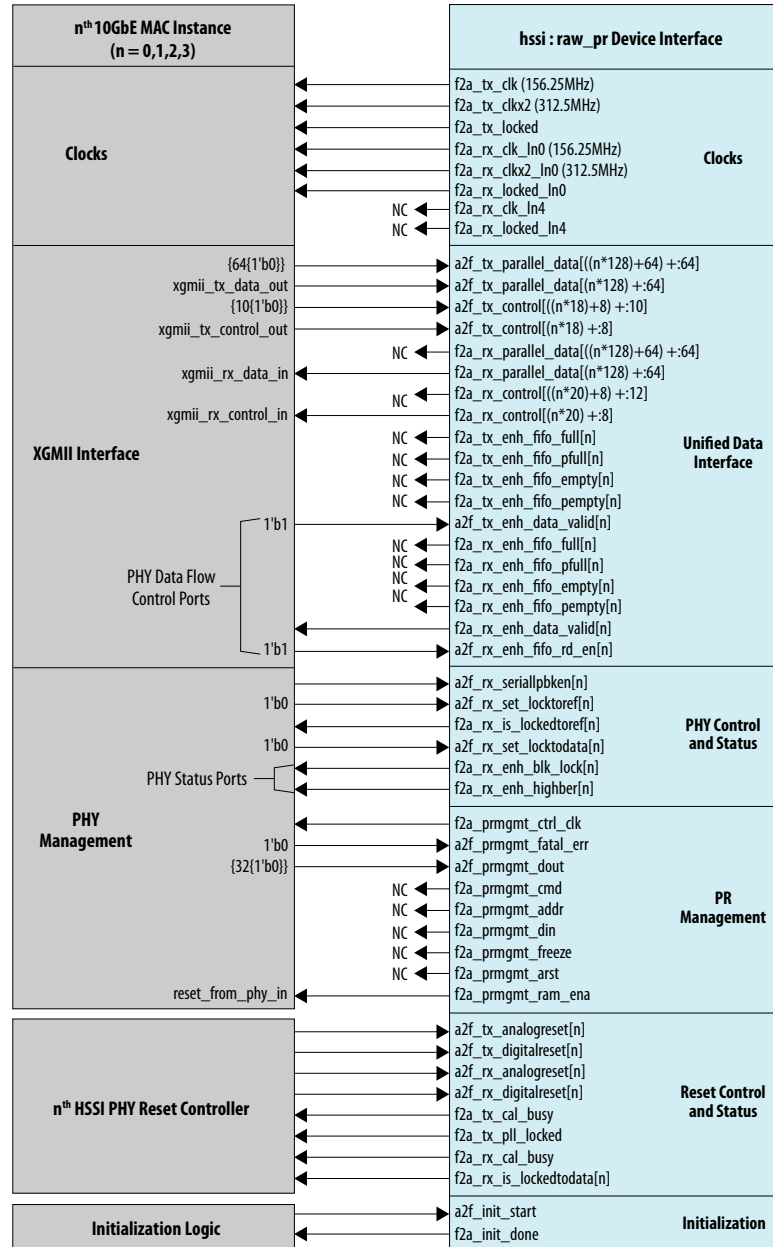
Related Information

- [10Gbps Ethernet Accelerator Functional Unit \(AFU\) Design Example User Guide](#)
- [40Gbps Ethernet Accelerator Functional Unit \(AFU\) Design Example User Guide](#)

2.2.1. 4x10GBASE-SR Mode

In 4x10GBASE-SR mode, the interface between the MAC and HSSI PHY maps to XGMII. The figure below and sections that follow describe how to connect 10GbE MAC IP to the HSSI PHY over the `hssi` interface.

Figure 3. Connection to HSSI PHY in 4x10GBASE-SR Mode



2.2.1.1. Clocks in 4x10GBASE-SR Mode

The hssi interface provides a set of clocks and locked status flags to support the 10GbE MAC IP. The interface provides clock sources of 156.25MHz and 312.5MHz for both transmit and receive datapaths. The XGMII interface between the MAC and HSSI PHY is synchronous to f2a_tx_clk and f2a_rx_clk_ln0 for transmit and receive, respectively. The 312.5MHz clock sources and locked status outputs from the fPLLs in the HSSI PHY can be used by the MAC and related AFU logic as needed.



2.2.1.2. Unified Data Interface in 4x10GBASE-SR Mode

Each 10GbE channel's XGMII data interface is striped across 128-bit segments of the unified data interface transmit and receive data ports. The 64-bit XGMII data is mapped to the lower 64 bits of the 128-bit segment. The upper 64 bits of the transmit datapath segment should be statically driven low. The upper 64 bits of the receive datapath segment should be left unconnected.

Each 10GbE channel's XGMII data control interface is striped across 18-bit segments of the unified data interface's transmit and receive data control ports. The eight bits of control are mapped to the lower eight bits of the 18-bit segment. The control bit for the least significant XGMII data byte lane (e.g., `xgmii_tx_data_out[7:0]`) maps to the least significant bit of the unified data interface's control port (e.g., `a2f_tx_parallel_control[0]` corresponds to `xgmii_tx_control_out[0]`) with each successive control bit mapping similarly to the same bit index of the unified data interface control port. The upper 10 bits of the transmit data control segment should be statically driven low. The upper 10 bits of the receive data control segment should be left unconnected.

The HSSI PHY FIFO flow control ports are not utilized in 4x10GBASE-SR mode. Statically drive the `a2f_tx_enh_data_valid` and `a2f_rx_enh_fifo_rd_en` ports high.

2.2.1.3. PHY Control and Status in 4x10GBASE-SR Mode

Statically drive `a2f_rx_set_locktoref` and `a2f_rx_set_locktodata` low to place the HSSI PHY receive PMA CDRs in auto-lock mode. The status outputs on these ports are available to the MAC and related AFU logic for optional use.

2.2.1.4. PR Management in 4x10GBASE-SR Mode

The `f2a_prmgmt_ctrl_clk` clock output is a 100MHz free running clock source that the MAC and related AFU logic can optionally use for miscellaneous lower speed logic. The MAC and related AFU logic can optionally use the `f2a_prmgmt_ram_ena` output as a reset.

The remaining ports on the PR management bus are for internal use in Intel AFU example designs. Statically drive `a2f_prmgmt_fatal_err` and `a2f_prmgmt_dout` low. In a multi-channel implementation, drive from a centralized point in the AFU implementation's top-level logic.

2.2.1.5. Reset Control and Status in 4x10GBASE-SR Mode

See the [Resetting Transceiver Channels](#) chapter in the *Intel Arria 10 Transceiver PHY User Guide* for details on using either the Intel Transceiver PHY Reset Controller IP included in Quartus Prime Pro or your own custom reset controller to properly sequence the resets for the serial transceiver blocks in the HSSI PHY. The *Connection to HSSI PHY in 4x10GBASE-SR Mode* figure in *4x10GBASE-SR Mode* shows the use of a separate reset controller per channel, but you could also implement a single centralized reset controller for all channel transceivers. The Intel Transceiver PHY Reset Controller IP can be configured for single or multi-channel use cases.

Related Information

- [4x10GBASE-SR Mode](#) on page 11
Figure: "Connection to HSSI PHY in 4x10GBASE-SR Mode"



- [Intel Arria 10 Transceiver PHY User Guide](#)
- [Resetting Transceiver Channels](#)
The *Resetting Transceiver Channels* chapter of the *Intel Arria 10 Transceiver PHY User Guide*

2.2.1.6. Initialization in 4x10GBASE-SR Mode

The MAC and related AFU logic can optionally use the handshake initialization signaling between the AFU and HSSI PHY. If you do not use the initialization handshake control, statically drive `a2f_init_start` high.

2.2.1.7. Unused 10GbE Channels

The 4x10GBASE-SR HSSI PHY mode supports one to four 10GbE channels. AFUs that implement less than four channels should terminate the unused channels by statically driving the `hssi` input ports listed in the below table to the levels shown.

Table 9. Terminating Unused 10GbE Channels

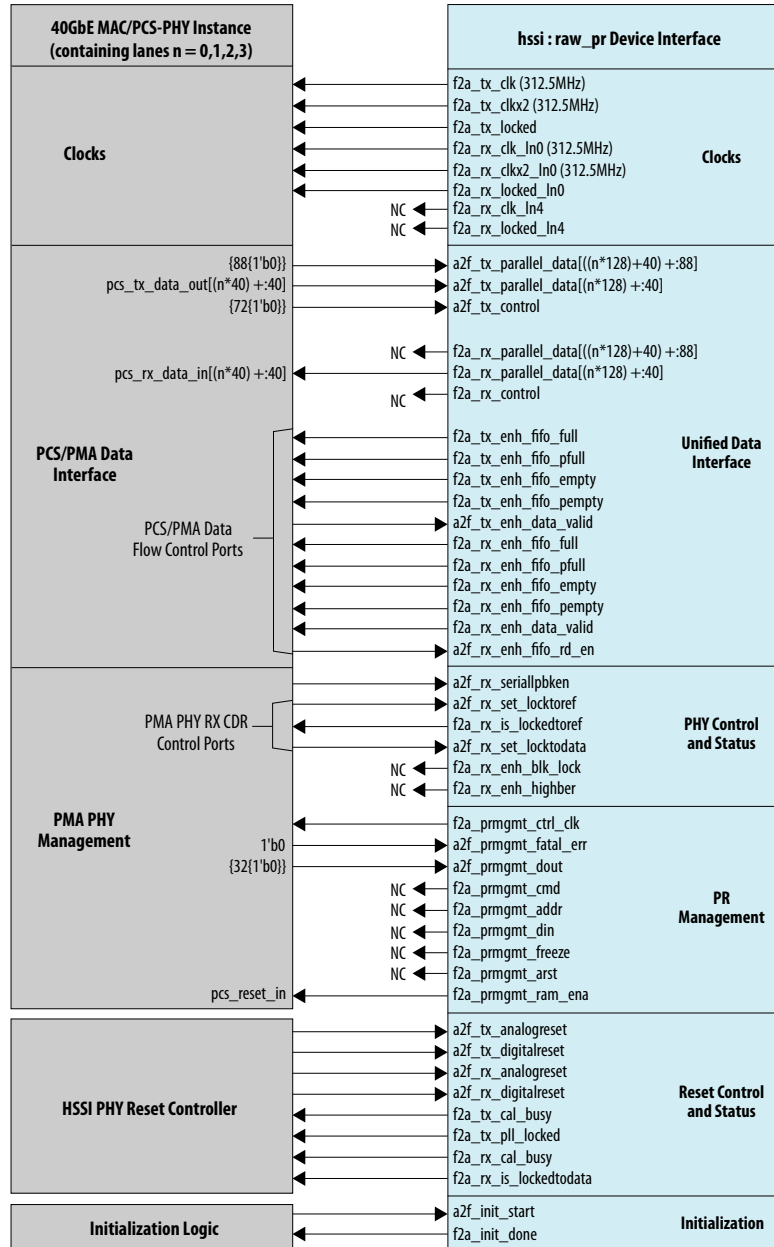
hssi Port Name on Unused Channel "n" (n = 0,1,2,3)	Port Termination Value
<code>a2f_tx_analogreset[n]</code>	1'b1
<code>a2f_tx_digitalreset[n]</code>	1'b1
<code>a2f_rx_analogreset[n]</code>	1'b1
<code>a2f_rx_digitalreset[n]</code>	1'b1
<code>a2f_rx_serialpbken[n]</code>	1'b1
<code>a2f_rx_set_locktodata[n]</code>	1'b0
<code>a2f_rx_set_locktoref[n]</code>	1'b0
<code>a2f_tx_enh_data_valid[n]</code>	1'b0
<code>a2f_rx_enh_fifo_rd_en[n]</code>	1'b0
<code>a2f_tx_parallel_data[(n*128) +:128]</code>	{128{1'b0}}
<code>a2f_tx_control[(n*18) +:18]</code>	{18{1'b0}}

2.2.2. 40GBASE-SR4 Mode

In 40GBASE-SR4 mode, the AFU must implement a 40GbE PCS layer between the 40GbE MAC and HSSI PHY configured for 40GBASE-SR4 PMA-only mode. The interface between the 40GbE PCS implemented in the AFU and the HSSI PMA PHY is a 40-bit transmit and receive interface with flow control. The figure below and sections that follow describe how to connect 40GbE MAC/PCS-PHY IP to the HSSI PMA PHY over the `hssi` interface.



Figure 4. Connection to HSSI PHY in 40GBASE-SR4 Mode



2.2.2.1. Clocks in 40GBASE-SR4 Mode

The `hssi` interface provides a set of clocks and locked status flags to support 40GbE MAC/PCS IP. The interface provides 312.5MHz clock sources for both transmit and receive datapaths. The 40-bit interface between the MAC/PHY and HSSI PHY is synchronous to `f2a_tx_clk` and `f2a_rx_clk_l1n0` for transmit and receive, respectively. The locked status outputs from the fPLLs in the HSSI PHY can be used by

the MAC and related AFU logic as needed. The additional set of 312.5MHz clock sources (e.g., `f2a_tx_clkx2`, `f2a_rx_clkx2_1n0`) are phase-aligned replications and would typically not be needed by the MAC/PHY and related AFU logic.

2.2.2.2. Unified Data Interface in 40GBASE-SR4 Mode

Each lane of the 40GbE PCS PHY data interface is striped across 128-bit segments of the unified data interface transmit and receive data ports. Each 40-bit PCS PHY data lane is mapped to the lower 40 bits of the 128-bit segment. The upper 88 bits of the transmit datapath segment should be statically driven low. The upper 88 bits of the receive datapath segment should be left unconnected.

The unified data interface's transmit and receive control ports are not utilized in 40GBASE-SR4 mode. Statically drive the `a2f_tx_control` port low and leave `f2a_rx_control` unconnected.

Flow control is utilized in 40GBASE-SR4 mode between the 40GbE PCS and HSSI PHY PMA layers. See the *HSSI Unified Data Interface* section for details about the flow control signaling.

Related Information

[HSSI Unified Data Interface](#) on page 7

2.2.2.3. PHY Control and Status in 40GBASE-SR4 Mode

Actively drive `a2f_rx_set_locktoref` and `a2f_rx_set_locktodata` and monitor `f2a_rx_is_lockedtoref` to control the HSSI PHY receive PMA CDRs lock sequence according to the *HSSI Unified Data Interface* section. The `f2a_rx_enh_blk_lock` and `f2a_rx_enh_highber` ports are not utilized in 40GBASE-SR4 mode – leave disconnected.

Related Information

[HSSI Unified Data Interface](#) on page 7

2.2.2.4. PR Management in 40GBASE-SR4 Mode

The `f2a_prmgmt_ctrl_clk` clock output is a 100MHz free running clock source that the MAC/PHY and related AFU logic can optionally use for miscellaneous lower speed logic. The 40GbE PCS layer should use the `f2a_prmgmt_ram_ena` output as a reset source.

The remaining ports on the PR management bus are for internal use in Intel AFU example designs. Statically drive `a2f_prmgmt_fatal_err` and `a2f_prmgmt_dout` low.

2.2.2.5. Reset Control and Status in 40GBASE-SR4 Mode

See the [Resetting Transceiver Channels](#) chapter in the *Intel Arria 10 Transceiver PHY User Guide* for details on using either the Intel Transceiver PHY Reset Controller IP included in Quartus Prime Pro or your own custom reset controller to properly sequence the resets for the serial transceiver blocks in the HSSI PHY. The above figure shows the use of a single controller for all transceiver lanes. The Intel Transceiver PHY Reset Controller IP can be configured for single or multi-lane use cases.



Related Information

[Resetting Transceiver Channels](#)

The *Resetting Transceiver Channels* chapter of the *Intel Arria 10 Transceiver PHY User Guide*

2.2.2.6. Initialization in 40GBASE-SR4 Mode

The MAC/PCS and related AFU logic can optionally use the handshake initialization signaling between the AFU and HSSI PHY. If you do not use the initialization handshake control, statically drive `a2f_init_start` high.

2.3. Verifying Network Port Function

The OPAE SDK does not support verifying network port functionality with the AFU Simulation Environment (ASE).

Intel recommends that you develop a standalone test harness to verify MAC-to-network port functionality with any of the following:

- Intel FPGA MAC/PHY IP
- Third-party IP
- Your proprietary IP

You can use the 10GbE or 40GbE sample AFU design as a starting point for your test harness.

The sample AFU designs use packet generation and monitoring blocks implemented in the AFU to facilitate loopback testing on the network port. The samples also include an OPAE test application with APIs to control testing and readback results on the host.

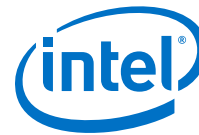
The following documents provide guidance on using the sample AFU designs as a template for standalone network port testing with your MAC/PHY connection to the `hssi` interface:

- [40Gbps Ethernet Accelerator Functional Unit \(AFU\) Design Example User Guide](#)
- [10Gbps Ethernet Accelerator Functional Unit \(AFU\) Design Example User Guide](#)

The user guides provide links to example AFU source code in the OPAE SDK.

Related Information

- [10Gbps Ethernet Accelerator Functional Unit \(AFU\) Design Example User Guide](#)
- [40Gbps Ethernet Accelerator Functional Unit \(AFU\) Design Example User Guide](#)



3. OPAE Support

The OPAE SDK includes the following support for the Intel PAC with Intel Arria 10 GX FPGA network port feature:

- OPAE kernel driver sysfs files enable configuration of the network port feature and access to related information on the Intel PAC with Intel Arria 10 GX FPGA from the host.
 - 128-bit UUID for the Intel PAC with Intel Arria 10 GX FPGA
 - Base MAC address
 - HSSI PHY mode configuration
 - HSSI PHY PMA analog settings
- Sample AFU designs for 10GbE and 40GbE

3.1. OPAE Driver HSSI sysfs Files

OPAE supports HSSI management on the host through driver sysfs files associated with the FME of each enumerated Intel PAC with Intel Arria 10 GX FPGA in the system under the following sysfs hierarchy:

```
/sys/class/fpga/intel-fpga-dev.<i>/intel-fpga-fme.<j>/intel-pac-hssi.<m>.auto/hssi_mgmt/
```

- Index *i* in `intel-fpga-dev.<i>` consecutively numbers all the enumerated FPGA container devices. In the Intel PAC with Intel Arria 10 GX FPGA hardware platform, each FPGA container device corresponds to a Intel PAC with Intel Arria 10 GX FPGA installed in the system.
- Index *j* in `intel-fpga-fme.<j>` consecutively numbers all the enumerated FMEs in the system.
- Index *m* in `intel-pac-hssi.<m>.auto` consecutively numbers all the enumerated HSSI controllers in the system.

3.1.1. The `board_id` sysfs

```
.../hssi_mgmt/board_id
```

The `board_id` sysfs is read-only and contains a 128-bit UUID associated with the Intel PAC with Intel Arria 10 GX FPGA. This is a board-level UUID, independent of any loaded AF UUID. The format of this file is the 128-bit UUID in binary.

3.1.2. The `eeeprom` sysfs

```
.../hssi_mgmt/eeeprom
```



The eeprom sysfs is read-only and contains ASCII-encoded fields delimited by LF (line feed, new line) characters as shown in the following table:

Table 10. EEPROM sysfs Fields

Field Description	Format (ASCII)
Base MAC Address	MAC=<hh>:<hh>:<hh>:<hh>:<hh>:<hh>\n
Board Serial Number	SN=<dddddd>\n
Board Information	PC=A10SA4-0U-B115X2E2Q-22-H501U0R-6\n
Board Revision	REV=<d>.<d>.<d>.<d>.\n

- Values enclosed in angle brackets (<>) are unique values per Intel PAC with Intel Arria 10 GX FPGA.
- A value of h enclosed in angle brackets indicates a unique hexadecimal digit
- A value of d indicates a unique decimal digit.

3.1.3. The config sysfs

```
.../hssi_mgmt/config
```

The config sysfs is write-only from a privileged process (root access). Write the appropriate string value to the config sysfs to configure the HSSI PHY mode as shown in the following table.

Table 11. config sysfs Values for HSSI PHY Mode

HSSI PHY Mode	config sysfs Write String Value
4x10GBASE-SR	10
40GBASE-SR4	40

3.1.4. The equalizer_tune sysfs

```
.../hssi_mgmt/equalizer_tune
```

The equalizer_tune sysfs is readable by all and writable from a privileged process (root access). It is formatted as a list of ASCII-encoded fields delimited by LF (line feed, new line) characters. Each field in the list contains a specific transceiver analog PMA setting for a specific HSSI PHY transceiver lane. There are four lanes, and each lane has a set of eight analog PMA settings. The format for each field in the list is as follows:

```
<hssi-phy-transceiver-lane-index>:<analog-pma-setting-index>=<analog-pma-setting>\n
```

The hssi-phy-transceiver-lane-index subfield specifies the HSSI PHY transceiver lane number as shown in the table below.

Table 12. HSSI PHY Transceiver Lane Index Subfield

Supported Range of ASCII-Encoded String Values	Description
"0"	HSSI PHY transceiver lane 0
"1"	HSSI PHY transceiver lane 1
"2"	HSSI PHY transceiver lane 2
"3"	HSSI PHY transceiver lane 3

The `analog-pma-setting-index` subfield specifies a transceiver analog PMA setting, or group of settings, for the HSSI PHY lane indicated by the `hssi-phy-transceiver-lane-index` subfield as shown in the table below.

Table 13. HSSI PHY Transceiver Analog PMA Setting Index Subfield

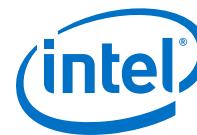
Supported Range of ASCII-Encoded String Values	Description	Reference in Intel Arria 10 Transceiver PHY User Guide
"0"	Receiver High Gain Mode Equalizer AC Gain Control (CTLE ACGAIN)	Table <i>Available Options</i> in XCVR_A10_RX_ADP_CTLE_ACGAIN_4S
"1"	Receiver Variable Gain Amplifier Voltage Swing Select (VGA)	Table <i>Available Options</i> in XCVR_A10_RX_ADP_VGA_SEL
"2"	Receiver High Gain Mode Equalizer DC Gain Control (CTLE DCGAIN)	Table <i>Available Options</i> in XCVR_A10_RX_EQ_DC_GAIN_TRIM
"3"	Transmitter Pre-Emphasis First Post-Tap Magnitude	Table <i>Available Options</i> in XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_1ST_POST_TAP
"4"	Transmitter Pre-Emphasis Second Post-Tap Magnitude	Table <i>Available Options</i> in XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_2ND_POST_TAP
"5"	Transmitter Pre-Emphasis First Pre-Tap Magnitude	Table <i>Available Options</i> in XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_PRE_TAP_1T
"6"	Transmitter Pre-Emphasis Second Pre-Tap Magnitude	Table <i>Available Options</i> in XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_PRE_TAP_2T
"7"	Transmitter Output Swing Level	Table <i>Available Options</i> in XCVR_A10_TX_VOD_OUTPUT_SWING_CTRL

The ASCII-encoded string value specified by the `analog-pma-setting` subfield corresponds to the encoded value for the HSSI PHY transceiver analog PMA setting on the lane indicated by the `hssi-phy-transceiver-lane-index:analog-pma-setting-index` subfields.

The `equalizer_tune` sysfs reads and configures analog PMA setting value encodings for the `analog-pma-setting` subfield. For descriptions of those setting value encodings, refer to the *Encodings* sections that follow.

Related Information

- [Receiver CTLE AC Gain sysfs Encodings](#) on page 21
- [Receiver VGA sysfs Encodings](#) on page 21



- [Receiver CTLE DC Gain sysfs Encodings](#) on page 21
- [Transmitter Pre-Emphasis First Post Tap Encodings](#) on page 22
- [Transmitter Pre-Emphasis Second Post Tap Encodings](#) on page 22
- [Transmitter Pre-Emphasis First Pre Tap Encodings](#) on page 23
- [Transmitter Pre-Emphasis Second Pre Tap Encodings](#) on page 23
- [Transmitter VOD Encodings](#) on page 24

3.1.4.1. Receiver CTLE AC Gain sysfs Encodings

```
analog-pma-setting-index = "0"
```

HSSI PHY receiver CTLE AC Gain is specified using the `XCVR_A10_RX_ADP_CTLE_ACGAIN_4S` parameter. The following table shows the supported range of values for receiver CTLE AC Gain with the corresponding sysfs `analog-pma-setting` hex string value.

Table 14. Receiver CTLE AC Gain sysfs Value Encodings

<code>XCVR_A10_RX_ADP_CTLE_ACGAIN_4S</code>	<code>analog-pma-setting</code>
<code>RADP_CTLE_ACGAIN_4S_<0 to 28></code>	Range of string values from "0" to "1c" (default = "0")

Related Information

The [equalizer_tune sysfs](#) on page 19

3.1.4.2. Receiver VGA sysfs Encodings

```
analog-pma-setting-index = "1"
```

HSSI PHY receiver Variable Gain Amplifier (VGA) voltage swing select is specified using the `XCVR_A10_RX_ADP_VGA_SEL` parameter. The following table shows the supported range of values for receiver VGA voltage swing with the corresponding sysfs `analog-pma-setting` hex string value.

Table 15. Receiver VGA sysfs Value Encodings

<code>XCVR_A10_RX_ADP_VGA_SEL</code>	<code>analog-pma-setting</code>
<code>RADP_VGA_SEL_<0 to 7></code>	Range of string values from 0 to 7 (default = 4)

Related Information

The [equalizer_tune sysfs](#) on page 19

3.1.4.3. Receiver CTLE DC Gain sysfs Encodings

```
analog-pma-setting-index = "2"
```

HSSI PHY receiver CTLE DC Gain is specified using the `XCVR_A10_RX_EQ_DC_GAIN_TRIM` parameter. The following table shows the supported range of values for receiver CTLE DC Gain with the corresponding sysfs `analog-pma-setting` hex string value.

Table 16. Receiver CTLE DC Gain sysfs Value Encodings

<code>XCVR_A10_RX_EQ_DC_GAIN_TRIM</code>	<code>analog-pma-setting</code>
<code>NO_DC_GAIN</code>	"0"
<code>STGI_GAIN7</code>	"7" (default)

Related Information

The `equalizer_tune sysfs` on page 19

3.1.4.4. Transmitter Pre-Emphasis First Post Tap Encodings

```
analog-pma-setting-index = "3"
```

HSSI PHY transmitter pre-emphasis first post tap is specified using a combination of two parameters:

- `XCVR_A10_TX_PRE_EMP_SIGN_1ST_POST_TAP` – specifies positive or negative pre-emphasis polarity.
- `XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_1ST_POST_TAP` – specifies pre-emphasis magnitude.

The following table shows the supported range of values for transmitter pre-emphasis first post tap with the corresponding sysfs `analog-pma-setting` hex string value.

Table 17. Transmitter Pre-Emphasis First Post Tap sysfs Value Encodings

<code>XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_1ST_POST_TAP</code>	<code>analog-pma-setting</code>
Range of decimal values from 0 to 25 <code>XCVR_A10_TX_PRE_EMP_SIGN_1ST_POST_TAP = FIR_POST_1T_POS</code>	Range of string values from "0" to "19" (default = "0")
Range of decimal values from 0 to 25 <code>XCVR_A10_TX_PRE_EMP_SIGN_1ST_POST_TAP = FIR_POST_1T_NEG</code>	Range of string values from "40" to "59"

Related Information

The `equalizer_tune sysfs` on page 19

3.1.4.5. Transmitter Pre-Emphasis Second Post Tap Encodings

```
analog-pma-setting-index = "4"
```

HSSI PHY transmitter pre-emphasis second post tap is specified using a combination of two parameters:

- `XCVR_A10_TX_PRE_EMP_SIGN_2ND_POST_TAP` – specifies positive or negative pre-emphasis polarity.
- `XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_2ND_POST_TAP` – specifies pre-emphasis magnitude.

The following table shows the supported range of values for transmitter pre-emphasis second post tap with the corresponding sysfs `analog-pma-setting` hex string value.



Table 18. Transmitter Pre-Emphasis Second Post Tap sysfs Value Encodings

<code>XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_2ND_POST_TAP</code>	<code>analog-pma-setting</code>
Range of decimal values from 0 to 12 <code>XCVR_A10_TX_PRE_EMP_SIGN_2ND_POST_TAP = FIR_POST_2T_POS</code>	Range of string values from "0" to "c"
Range of decimal values from 0 to 12 <code>XCVR_A10_TX_PRE_EMP_SIGN_2ND_POST_TAP = FIR_POST_2T_NEG</code>	Range of string values from "20" to "2c" (default = "20")

Related Information

The `equalizer_tune sysfs` on page 19

3.1.4.6. Transmitter Pre-Emphasis First Pre Tap Encodings

```
analog-pma-setting-index = "5"
```

HSSI PHY transmitter pre-emphasis first pre tap is specified using a combination of two parameters:

- `XCVR_A10_TX_PRE_EMP_SIGN_PRE_TAP_1T` – specifies positive or negative pre-emphasis polarity.
- `XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_PRE_TAP_1T` – specifies pre-emphasis magnitude.

The following table shows the supported range of values for transmitter pre-emphasis first pre tap with the corresponding sysfs `analog-pma-setting` hex string value.

Table 19. Transmitter Pre-Emphasis First Pre Tap sysfs Value Encodings

<code>XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_PRE_TAP_1T</code>	<code>analog-pma-setting</code>
Range of decimal values from 0 to 16 <code>XCVR_A10_TX_PRE_EMP_SIGN_PRE_TAP_1T=FIR_PRE_1T_POS</code>	Range of string values from "0" to "10"
Range of decimal values from 0 to 16 <code>XCVR_A10_TX_PRE_EMP_SIGN_PRE_TAP_1T=FIR_PRE_1T_NEG</code>	Range of string values from "20" to "30" (default = "20")

Related Information

The `equalizer_tune sysfs` on page 19

3.1.4.7. Transmitter Pre-Emphasis Second Pre Tap Encodings

```
analog-pma-setting-index = "6"
```

HSSI PHY transmitter pre-emphasis second pre tap is specified using a combination of two parameters:

- `XCVR_A10_TX_PRE_EMP_SIGN_PRE_TAP_2T` – specifies positive or negative pre-emphasis polarity.
- `XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_PRE_TAP_2T` – specifies pre-emphasis magnitude.

The following table shows the supported range of values for transmitter pre-emphasis second pre tap with the corresponding sysfs `analog-pma-setting` hex string value.

Table 20. Transmitter Pre-Emphasis Second Pre Tap sysfs Value Encodings

<code>XCVR_A10_TX_PRE_EMP_SWITCHING_CTRL_PRE_TAP_2T</code>	<code>analog-pma-setting</code>
Range of decimal values from 0 to 7 <code>XCVR_A10_TX_PRE_EMP_SIGN_PRE_TAP_2T=FIR_PRE_2T_POS</code>	Range of string values from "0" to "7"
Range of decimal values from 0 to 7 <code>XCVR_A10_TX_PRE_EMP_SIGN_PRE_TAP_2T=FIR_PRE_2T_NEG</code>	Range of string values from "10" to "17" (default = "10")

Related Information

The `equalizer_tune sysfs` on page 19

3.1.4.8. Transmitter VOD Encodings

```
analog-pma-setting-index = "7"
```

HSSI PHY transmitter VOD is specified using the `XCVR_A10_TX_VOD_OUTPUT_SWING_CTRL` parameter. The following table shows the supported range of values for transmitter VOD with the corresponding sysfs `analog-pma-setting` hex string value.

Table 21. Transmitter VOD Encodings

<code>XCVR_A10_TX_VOD_OUTPUT_SWING_CTRL</code>	<code>analog-pma-setting</code>
Range of decimal values from 0 to 31	Range of string values from "0" to "1f" (default = "1d")

Related Information

The `equalizer_tune sysfs` on page 19

3.2. Managing the Network Port

Manage the network port from the host using the OPAE driver sysfs files. The sections that follow cover some management use cases. You must read *Configuring the Network Port* before using the feature from any host application.

The examples illustrate managing the network port through the driver sysfs files from a shell terminal window using Linux commands and OPAE tools. From a host application, use calls to `open()`, `read()`, `write()` and `close()` from the `stdio` library on the sysfs files and parse the strings according to the format documented in the preceding sections.

Related Information

Configuring the Network Port on page 24

3.2.1. Configuring the Network Port

To enable the network port, you must configure the HSSI PHY mode and load a network port-enabled AF from the host. The following procedure shows the programming method for 4x10GBASE-SR operation from a shell terminal window using Linux commands and OPAE tools for a single Intel PAC with Intel Arria 10 GX FPGA installed in the system.



1. Configure the HSSI PHY on the Intel PAC with Intel Arria 10 GX FPGA using the driver config sysfs file.

```
$ sudo sh -c "echo 10 > \  
/sys/class/fpga/intel-fpga-dev.<i>/intel-fpga-fme.<j>/intel-pac-  
hssi.<m>.auto/hssi_mgmt/config"
```

2. Load an AF that supports the configured HSSI PHY mode.

```
$ sudo fpgaconfig \  
$OPAE_PLATFORM_ROOT/hw/samples/eth_e2e_e10/bin/eth_e2e_e10.gbs
```

After performing the above steps, the network port on the Intel PAC with Intel Arria 10 GX FPGA is ready for OPAE applications compatible with the loaded AF.

3.2.2. Reading the Base MAC Address from the Intel PAC with Intel Arria 10 GX FPGA

Each Intel PAC with Intel Arria 10 GX FPGA reserves four consecutive MAC addresses. The Intel PAC with Intel Arria 10 GX FPGA stores a single, universally unique base MAC address. For 4x10GBASE-SR mode, the Intel PAC with Intel Arria 10 GX FPGA reserves the next three consecutive addresses.

Read the kernel driver `eeprom` sysfs file to retrieve the base MAC address as follows:

```
$ hexdump -C \  
/sys/class/fpga/intel-fpga-dev.<i>/intel-fpga-fme.<j>/intel-pac-hssi.<m>.auto/  
hssi_mgmt/eeprom
```

The fields are delimited by LF (new line) characters. The base MAC address is located in the first field.

3.2.3. Modifying HSSI PHY Transceiver PMA Settings

You can retrieve the current set of HSSI PHY transceiver analog PMA settings and modify individual settings per transceiver lane using the `equalizer_tune` sysfs file.

The following command dumps the current settings to `stdout`:

```
$ cat /sys/class/fpga/intel-fpga-dev.<i>/intel-fpga-fme.<j>/intel-pac-  
hssi.<m>.auto/hssi_mgmt/  
equalizer_tune
```

See *The equalizer_tune sysfs* for details on the format of this sysfs file's contents.

To modify an analog PMA setting for a transceiver lane, write a single field at a time to the `equalizer_tune` sysfs file. For example, to set the transmitter pre-emphasis second pre-tap magnitude to 4 for transceiver lane 2, write the following string value:

```
"2:6=4"
```

Here is an example of doing this from a shell terminal window:

```
$ sudo sh -c "echo 2:6=4 > \  
/sys/class/fpga/intel-fpga-dev.<i>/intel-fpga-fme.<j>/intel-pac-hssi.<m>.auto/  
hssi_mgmt/equalizer_tune"
```

Do this separately for each lane or PMA setting value you want to modify.



Related Information

The `equalizer_tune sysfs` on page 19

3.2.3.1. HSSI PHY Analog PMA Presets

The HSSI PHY's analog PMA settings must be configured for the specific QSFP+ interconnect media used to connect the Intel PAC with Intel Arria 10 GX FPGA to the network. See the *Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA Datasheet* for the procedure for applying analog PMA presets for the supported interconnect media.

Related Information

[Intel® Programmable Acceleration Card \(PAC\) with Intel® Arria® 10 GX FPGA Datasheet](#)



4. Networking Interface for Open Programmable Acceleration Engine: Intel PAC with Intel Arria 10 GX FPGA Revision History

Table 22. Networking Interface for Open Programmable Acceleration Engine: Intel PAC with Intel Arria 10 GX FPGA Revision History

Document Version	Changes
2019.08.05	Changed the document title to Networking Interface for Open Programmable Acceleration Engine: Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA . Previously titled as <i>HSSI User Guide for Intel Programmable Acceleration Card (PAC) Intel Arria 10 GX FPGA</i> .
2018.12.04	Maintenance release. This document supports the OPAE SDK versions 1.1 and 1.2.
2018.08.06	Initial release.

Intel Corporation. All rights reserved. Agilix, Altera, Arria, Cyclone, Enpirion, Intel, the Intel logo, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered