
Ettus USRP X410 User Manual

2025-07-23



Contents

Ettus USRP X410 Block Diagrams	3
Socketed CLIP Interface.....	4
QSFP28 0 and QSFP28 1 Quad Mapping.....	4
QSFP28 Sockets	5
Supported Aurora CLIPs	13
Supported 10 GbE CLIPs.....	15
Using the Ettus USRP X410 with the LabVIEW FPGA Module.....	21

Ettus USRP X410 Block Diagrams

Signals received by the Ettus USRP X410 are amplified, downconverted, filtered, digitized, and decimated before being passed to the host computer. Signals transmitted by the Ettus USRP X410 are upsampled, reconstructed, filtered, upconverted, and amplified before being transmitted.

Figure 1. Ettus USRP X410 Motherboard Block Diagram

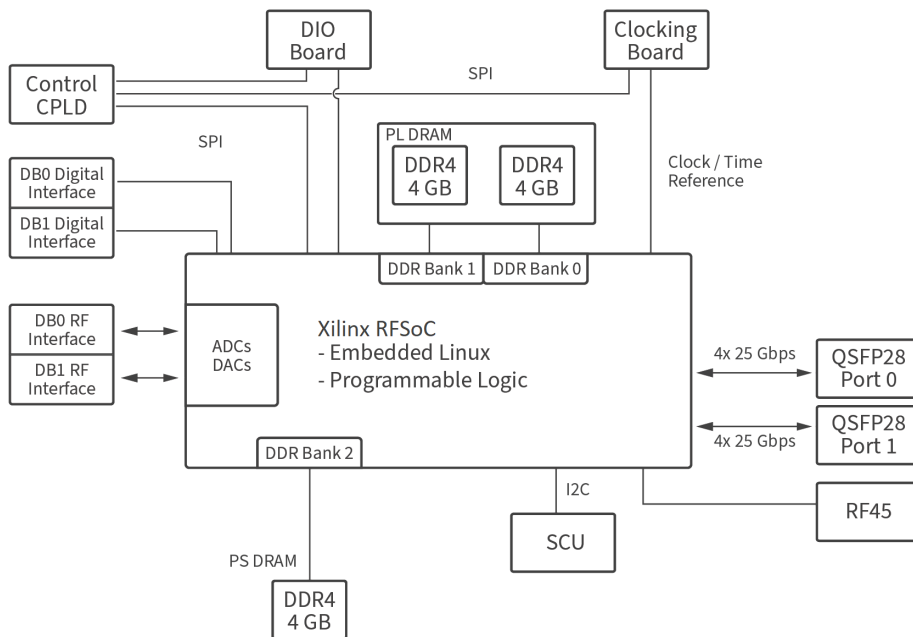
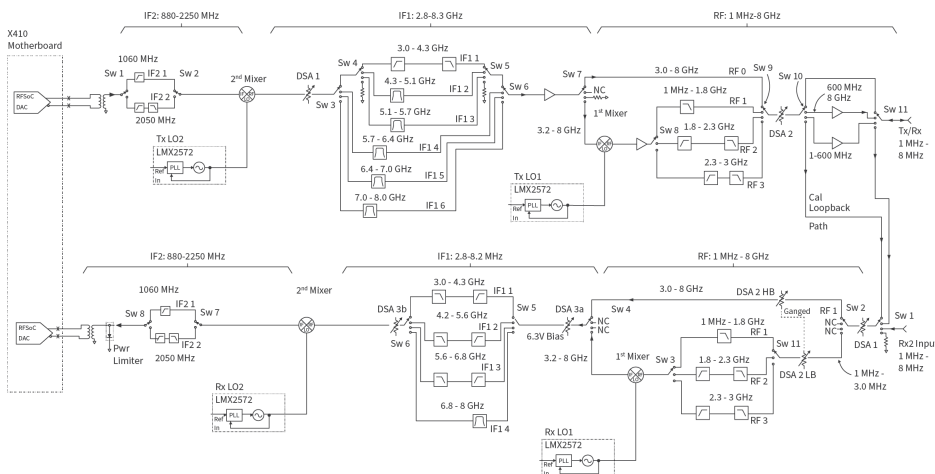


Figure 1. Ettus USRP X410 ZBX Block Diagram



Socketed CLIP Interface

The Ettus USRP X410 socketed CLIP allows you to insert HDL IP into an FPGA target, enabling VHDL code to communicate directly with an FPGA VI. Socketed CLIP also allows the CLIP to communicate directly with circuitry external to the FPGA. The following sections provide information about how to configure your device for use with the Ettus USRP X410 socketed CLIP.

QSFP28 0 and QSFP28 1 Quad Mapping

The Ettus USRP X410 socketed CLIP uses the QSFP28 0 and QSFP28 1 ports on the front panel. Refer to the following tables for an overview of how these connectors map to resources.

Table 1. Transceiver Lane and Quad Mapping

Connector	Lane	Quad Location	Physical Resource
QSFP28 0	0	Quad 131	GTYE4_X0Y16
	1		GTYE4_X0Y17
	2		GTYE4_X0Y18
	3		GTYE4_X0Y19
QSFP28 1	0	Quad 128	GTYE4_X0Y4
	1		GTYE4_X0Y5
	2		GTYE4_X0Y6
	3		GTYE4_X0Y7

Table 2. Clock Signal and Quad Mapping

Connector	Reference Clock Option	Frequency	Quad Location	Physical Resource
QSFP28 0	1	156.25 MHz	Quad 130	MGTREFCLK0_130
	2	100 MHz		MGTREFCLK0_130
QSFP28 1	1	156.25 MHz	Quad 129	MGTREFCLK0_129

Connector	Reference Clock Option	Frequency	Quad Location	Physical Resource
	2	100 MHz		MGTREFCLK0_129

QSFP28 Sockets

Refer to the following tables for an overview of the Ettus USRP X410 QSFP28 sockets.

Table 3. Socket Required Signals (Basic)

Port	Direction	Clock Domain	Type	Description
QSFP<0,1>_RefClk1	In	—	std_logic	Single-ended input clock (reference clock option 1) configured to run at 156.25 MHz.
QSFP<0,1>_RefClk2	In	—	std_logic	Single-ended input clock (reference clock option 1) configured to run at 100 MHz.
QSFP<0,1>_RX_P	In (pad)	—	std_logic_vector (3 downto 0)	Dedicated differential MGT receives signals for all lanes <0...3> of QSFP28<0,1>.
QSFP<0,1>_RX_N	In (pad)	—	std_logic_vector (3 downto 0)	
QSFP<0,1>_TX_P	Out (pad)	—	std_logic_vector (3 downto 0)	Dedicated differential MGT receives signals for all lanes <0...3> of QSFP28<0,1>.
QSFP<0,1>_TX_N	Out (pad)	—	std_logic_vector (3 downto 0)	
QSFP<0,1>_ModPrs_n	In	Async	std_logic	Active-low

Port	Direction	Clock Domain	Type	Description
				module present signal from QSFP28 connector. This signal is asserted when the module for the port <0,1> is absent. This signal is grounded when a module is connected. This signal is pulled asserted when no module is present.
QSFP<0,1>_Reset_n	Out	Async	std_logic	Active-low reset signal to QSFP28<0,1>.
QSFP<0,1>_LPMode_n	Out	Async	std_logic	Active-low power mode control to QSFP28<0,1>.
QSFP<0,1>_LED_Active	Out	Async	std_logic_vector (3 downto 0)	The corresponding LED (Red) for QSFP28<0,1> turns on when this signal is driven high. It is recommended to drive this

Port	Direction	Clock Domain	Type	Description
				signal using a pulse stretcher to allow visual perception. You can drive this signal asynchronously if you provide a 40 ns minimum assertion time.
QSFP<0,1>_LED_Present	Out	Async	std_logic_vector (3 downto 0)	The corresponding LED (Green) for QSFP28 <0,1> turns on when this signal is driven high. It is recommended to drive this signal using a pulse stretcher to allow visual perception. You can drive this signal asynchronously if you provide a 40 ns minimum assertion time.
QSFP<0,1>_SysClk	In	—	std_logic	Fixed, single-ended, free-running 100 MHz clock. Primary

Port	Direction	Clock Domain	Type	Description
				purpose is for DRP configuration and/or to register and debounce the PMA initialization.

Table 4. Socket Required Signals (Advanced)

Port	Direction	Clock Domain	Type	Description
RxRecClkOut<0,1>	Out	—	std_logic	Single-ended output clock recovered from MGT. This signal is intended for clock recovery applications that drive the recovered clock to the device's clocking aux. board through a OBUFDS_GTE4 output buffer.
Clk40MhzQsfp<0,1>Socket	In	—	std_logic	Fixed, single- ended, free- running 40 MHz clock. This clock is used for memory- mapped communication with the Processing System AXI4-Lite interface.

Port	Direction	Clock Domain	Type	Description
cAxiAppQsfp<0,1>Awaddr	In	Clk40MhzQsfp<0,1>Socket	std_logic_vector(39 downto 0)	AXI4-Lite interface to enable low-throughput communication initiating from the on-board ARM (RFSoc's Processing System) to the IP in the QSFP28 <0,1> socketed CLIP.
cAxiAppQsfp<0,1>Awprot	In	Clk40MhzQsfp<0,1>Socket	std_logic_vector(2 downto 0)	
cAxiAppQsfp<0,1>Awvalid	In	Clk40MhzQsfp<0,1>Socket	std_logic	
cAxiAppQsfp<0,1>Awready	Out	Clk40MhzQsfp<0,1>Socket	std_logic	
cAxiAppQsfp<0,1>Wdata	In	Clk40MhzQsfp<0,1>Socket	std_logic_vector(31 downto 0)	
cAxiAppQsfp<0,1>Wstrb	In	Clk40MhzQsfp<0,1>Socket	std_logic_vector(3 downto 0)	
cAxiAppQsfp<0,1>Wvalid	In	Clk40MhzQsfp<0,1>Socket	std_logic	This interface is connected through PL AXI interconnects to the Processing System's M_AXI_HPM0_FPD master interface.
cAxiAppQsfp<0,1>Wready	Out	Clk40MhzQsfp<0,1>Socket	std_logic	
cAxiAppQsfp<0,1>Bresp	Out	Clk40MhzQsfp<0,1>Socket	std_logic_vector(1 downto 0)	
cAxiAppQsfp<0,1>Bvalid	Out	Clk40MhzQsfp<0,1>Socket	std_logic	
cAxiAppQsfp<0,1>Bready	In	Clk40MhzQsfp<0,1>Socket	std_logic	The register space reserved in the Processing System for the QSFP28<0,1> sockets are as follows:
cAxiAppQsfp<0,1>Araddr	In	Clk40MhzQsfp<0,1>Socket	std_logic_vector(39 downto 0)	
cAxiAppQsfp<0,1>Arprot	In	Clk40MhzQsfp<0,1>Socket	std_logic_vector(2 downto 0)	
cAxiAppQsfp<0,1>Arvalid	In	Clk40MhzQsfp<0,1>Socket	std_logic	
cAxiAppQsfp<0,1>Arready	Out	Clk40MhzQsfp<0,1>Socket	std_logic	QSFP28 0
cAxiAppQsfp<0,1>Rdata	Out	Clk40MhzQsfp<0,1>Socket	std_logic_vector(31 downto 0)	
cAxiAppQsfp<0,1>Rresp	Out	Clk40MhzQsfp<0,1>Socket	std_logic_vector(1 downto 0)	Base address: 0x1310000000
cAxiAppQsfp<0,1>Rvalid	Out	Clk40MhzQsfp<0,1>Socket	std_logic	Range: 16M (0x1000000)
cAxiAppQsfp<0,1>Rready	In	Clk40MhzQsfp<0,1>Socket	std_logic	QSFP28 1

Port	Direction	Clock Domain	Type	Description
				Base address: 0x1311000000 Range: 16M (0x1000000)
XBarClkQsfp<0,1>Socket	In	N/A	std_logic	Fixed, single-ended, free-running 200 MHz clock. This clock is used for memory-mapped communication with the Processing System AXI4-Full interface.
xSAxiQsfp<0,1>Awid	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(5 downto 0)	AXI4-Full interface to enable high-throughput communication initiating from the IP in the QSFP28 <0,1> socketed CLIP to the on-board ARM (RFSoc's Processing System). This interface is connected, through PL AXI interconnects, to
xSAxiQsfp<0,1>Awaddr	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(48 downto 0)	
xSAxiQsfp<0,1>Awlen	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(7 downto 0)	
xSAxiQsfp<0,1>Awsz	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(2 downto 0)	
xSAxiQsfp<0,1>Awburst	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(1 downto 0)	
xSAxiQsfp<0,1>Awlock	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(0 downto 0)	
xSAxiQsfp<0,1>Awcache	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(3 downto 0)	
xSAxiQsfp<0,1>Awprot	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(2	

Port	Direction	Clock Domain	Type	Description
			downto 0)	the Processing System's S_AXI_HPC1_FPD slave cache-coherent interface. This interface provides full access to the Processing System's memory space. Therefore, special care should be taken when utilizing this interface to initiate communication requests to the Processing System.
xSAxiQsfp<0,1>Awqos	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(3 downto 0)	
xSAxiQsfp<0,1>Awvalid	Out	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Awready	In	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Wdata	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(127 downto 0)	
xSAxiQsfp<0,1>Wstrb	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(15 downto 0)	
xSAxiQsfp<0,1>Wlast	Out	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Wvalid	Out	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Wready	In	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Bid	In	XBarClkQsfp<0,1>Socket	std_logic_vector(5 downto 0)	
xSAxiQsfp<0,1>Bresp	In	XBarClkQsfp<0,1>Socket	std_logic_vector(1 downto 0)	
xSAxiQsfp<0,1>Bvalid	In	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Bready	Out	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Arid	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(5 downto 0)	
xSAxiQsfp<0,1>Araddr	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(48 downto 0)	
xSAxiQsfp<0,1>Arlen	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(7 downto 0)	
xSAxiQsfp<0,1>Arsz	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(2 downto 0)	
xSAxiQsfp<0,1>Arburst	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(1 downto 0)	
xSAxiQsfp<0,1>Arlock	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(0 downto 0)	
xSAxiQsfp<0,1>Arcache	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(3	

Port	Direction	Clock Domain	Type	Description
			downto 0)	
xSAxiQsfp<0,1>Arprot	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(2 downto 0)	
xSAxiQsfp<0,1>Arqos	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(3 downto 0)	
xSAxiQsfp<0,1>Arvalid	Out	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Arready	In	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Rid	In	XBarClkQsfp<0,1>Socket	std_logic_vector(5 downto 0)	
xSAxiQsfp<0,1>Rdata	In	XBarClkQsfp<0,1>Socket	std_logic_vector(127 downto 0)	
xSAxiQsfp<0,1>Rresp	In	XBarClkQsfp<0,1>Socket	std_logic_vector(1 downto 0)	
xSAxiQsfp<0,1>Rlast	In	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Rvalid	In	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Rready	Out	XBarClkQsfp<0,1>Socket	std_logic	
xSAxiQsfp<0,1>Aruser	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(0 downto 0)	
xSAxiQsfp<0,1>Awuser	Out	XBarClkQsfp<0,1>Socket	std_logic_vector(0 downto 0)	
aQsfp<0,1>Iirq0	Out	Async	std_logic	Interrupts signals to the Processing System to be used by the CLIP's IP. These ports are directly connected to the PL IRQ1 vector, which is an interrupt source for the Generic Interrupt
aQsfp<0,1>Iirq1	Out	Async	std_logic	

Port	Direction	Clock Domain	Type	Description
				Controller (GIC) in the Processing System. QSFP28 0 aQsfp0Irq0: GIC IRQ1 Bit 4 aQsfp0Irq1: GIC IRQ1 Bit 5 QSFP28 1 aQsfp1Irq0: GIC IRQ1 Bit 6 aQsfp1Irq1: GIC IRQ1 Bit 7
kQsfp<0,1>VersionInfo	Out	Constant	std_logic_vector(95 downto 0)	A constant vector that is used internally to track the version of CLIPs provided with Ettus USRP X410 support. You can drive this vector with zeros.

Supported Aurora CLIPs

The Aurora CLIP shipped with the NI-USRP installer may be used with QSFP28 sockets in the Ettus USRP X410 target, each socket (0, 1) having its corresponding CLIP. Additionally, there are two different Aurora streaming rates supported, 12.5 Gbps and

25 Gbps.

The NI-USRP driver support provides Aurora examples (project templates) for the user to get started.

Installer location: <LabVIEW Directory>\Targets\NI\FPGA\USRP\X410\CLIP\AuroraCLIP\

Socket	Aurora 12.5 Gbps Streaming Rate	Aurora 25 Gbps Streaming Rate
QSFP28 0 Socket	AuroraClip_12.5Gbps_Port0.xml	AuroraClip_25Gbps_Port0.xml
QSFP28 1 Socket	AuroraClip_12.5Gbps_Port1.xml	AuroraClip_25Gbps_Port1.xml

The following table documents all the user-facing CLIP I/O nodes available to the user when populating one of the QSFP28 sockets with either version (rate) of the Aurora CLIPs.

Table 5. Aurora CLIP (12.5 Gbps or 25 Gbps) Signals

Port	Direction	Clock Domain	Type
UserQsfp<0,1>Clk	Out	—	Clock
InitQsfp<0,1>Clk	Out	—	Clock
uQsfp<0,1>TxTData<0...3>	In	UserQsfp<0,1>Clk	Unsigned 64-bit
uQsfp<0,1>TxTValid	In	UserQsfp<0,1>Clk	Boolean
uQsfp<0,1>TxTReady	Out	UserQsfp<0,1>Clk	Boolean
uQsfp<0,1>RxTData<0...3>	Out	UserQsfp<0,1>Clk	Unsigned 64-bit
uQsfp<0,1>RxTValid	Out	UserQsfp<0,1>Clk	Boolean
uQsfp<0,1>HardError	Out	UserQsfp<0,1>Clk	Boolean
uQsfp<0,1>SoftError	Out	UserQsfp<0,1>Clk	Boolean
uQsfp<0,1>LaneUp	Out	UserQsfp<0,1>Clk	Unsigned 4-bit
uQsfp<0,1>ChannelUp	Out	UserQsfp<0,1>Clk	Boolean
uQsfp<0,1>SysResetOut	Out	UserQsfp<0,1>Clk	Boolean
iQsfp<0,1>LinkResetOut	Out	InitQsfp<0,1>Clk	Boolean

Port	Direction	Clock Domain	Type
iQsfp<0,1>CoreReset	In	InitQsfp<0,1>Clk	Boolean
iQsfp<0,1>DrpAWAddr	In	InitQsfp<0,1>Clk	Unsigned 32-bit
iQsfp<0,1>DrpAWValid	In	InitQsfp<0,1>Clk	Boolean
iQsfp<0,1>DrpAWReady	Out	InitQsfp<0,1>Clk	Boolean
iQsfp<0,1>DrpWStrb	In	InitQsfp<0,1>Clk	Unsigned 4-bit
iQsfp<0,1>DrpWData	In	InitQsfp<0,1>Clk	Unsigned 32-bit
iQsfp<0,1>DrpWValid	In	InitQsfp<0,1>Clk	Boolean
iQsfp<0,1>DrpWReady	Out	InitQsfp<0,1>Clk	Boolean
iQsfp<0,1>DrpBValid	Out	InitQsfp<0,1>Clk	Boolean
iQsfp<0,1>DrpBReady	In	InitQsfp<0,1>Clk	Boolean
iQsfp<0,1>DrpBResp	Out	InitQsfp<0,1>Clk	Unsigned 2-bit
iQsfp<0,1>DrpARAddr	In	InitQsfp<0,1>Clk	Unsigned 32-bit
iQsfp<0,1>DrpARValid	In	InitQsfp<0,1>Clk	Boolean
iQsfp<0,1>DrpARReady	Out	InitQsfp<0,1>Clk	Boolean
iQsfp<0,1>DrpRData	Out	InitQsfp<0,1>Clk	Unsigned 32-bit
iQsfp<0,1>DrpRResp	Out	InitQsfp<0,1>Clk	Unsigned 2-bit
iQsfp<0,1>DrpRValid	Out	InitQsfp<0,1>Clk	Boolean
iQsfp<0,1>DrpRReady	In	InitQsfp<0,1>Clk	Boolean

Supported 10 GbE CLIPs

The 10 GbE CLIP shipped with the NI-USRP installer can be used with QSFP28 sockets in the Ettus USRP X410 target. Each socket (0, 1) has a corresponding CLIP.

Additionally, there are two different variants supported, 1 x 10 GbE (lane 0 has a 10 GbE connection, other lanes in the connector are unused) or 4 x 10 GbE (each lane in the QSFP28 connector has a 10 GbE connection).

NI-USRP driver support provides 10 GbE examples (project templates) for the user to get started.

Installer location: <LabVIEW Directory>\Targets\NI\FPGA\USRP\X410\CLIP\
TenGigEth_CLIP\

Socket	1 x 10 GbE Variant	4 x 10 GbE Variant
QSFP28 0 Socket	TenGigEth_x1_ClipPort0.xml	TenGigEth_x4_ClipPort0.xml
QSFP28 1 Socket	TenGigEth_x1_ClipPort1.xml	TenGigEth_x4_ClipPort1.xml

Documents all the user-facing CLIP I/O nodes available to the user when populating one of the QSFP28 sockets with either version of the 10 GbE CLIPs. Note that the 1 x 10 GbE CLIP only supports a single lane.

Table 6. 10 GbE CLIP (1 x 10 GbE or 4 x 10 GbE) Signals

Port	Direction	Clock Domain	Type	Description
UserQsfp<0,1>Clk	Out	—	Clock	AXI-Stream User Clock output. The frequency for this clock is 156.25 MHz.
uQsfp<0,1>Lane0TxTData	In	UserQsfp<0,1>Clk	Unsigned 64-bit	AXI-Stream TX interface, data vectors to core. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1TxTData	In	UserQsfp<0,1>Clk	Unsigned 64-bit	
uQsfp<0,1>Lane2TxTData	In	UserQsfp<0,1>Clk	Unsigned 64-bit	
uQsfp<0,1>Lane3TxTData	In	UserQsfp<0,1>Clk	Unsigned 64-bit	
uQsfp<0,1>Lane0TxTValid	In	UserQsfp<0,1>Clk	Boolean	AXI-Stream TX interface, data valid inputs to core. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1TxTValid	In	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane2TxTValid	In	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane3TxTValid	In	UserQsfp<0,1>Clk	Boolean	

Port	Direction	Clock Domain	Type	Description
uQsfp<0,1>Lane0TxTLast	In	UserQsfp<0,1>Clk	Boolean	AXI-Stream TX interface, end of Ethernet packet. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1TxTLast	In	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane2TxTLast	In	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane3TxTLast	In	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane0TxTUser	In	UserQsfp<0,1>Clk	Boolean	AXI-Stream TX interface, signals used to signal explicit underrun. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1TxTUser	In	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane2TxTUser	In	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane3TxTUser	In	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane0TxTKeep	In	UserQsfp<0,1>Clk	Unsigned 8-bit	AXI-Stream TX interface, data control to the core. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1TxTKeep	In	UserQsfp<0,1>Clk	Unsigned 8-bit	
uQsfp<0,1>Lane2TxTKeep	In	UserQsfp<0,1>Clk	Unsigned 8-bit	
uQsfp<0,1>Lane3TxTKeep	In	UserQsfp<0,1>Clk	Unsigned 8-bit	
uQsfp<0,1>Lane0TxTReady	Out	UserQsfp<0,1>Clk	Boolean	AXI-Stream TX interface, acknowledge signals from core to indicate the start of a data transfer. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1TxTReady	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane2TxTReady	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane3TxTReady	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane0RxTData	Out	UserQsfp<0,1>Clk	Unsigned 64-bit	AXI-Stream RX interface, data from core to upper layer. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1RxTData	Out	UserQsfp<0,1>Clk	Unsigned 64-bit	
uQsfp<0,1>Lane2RxTData	Out	UserQsfp<0,1>Clk	Unsigned 64-bit	

Port	Direction	Clock Domain	Type	Description
uQsfp<0,1>Lane3RxTData	Out	UserQsfp<0,1>Clk	Unsigned 64-bit	available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane0RxTValid	Out	UserQsfp<0,1>Clk	Boolean	AXI-Stream RX interface, data valid from core. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1RxTValid	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane2RxTValid	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane3RxTValid	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane0RxTLast	Out	UserQsfp<0,1>Clk	Boolean	AXI-Stream RX interface, signals from core indicating the end of a packet. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1RxTLast	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane2RxTLast	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane3RxTLast	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane0RxTUser	Out	UserQsfp<0,1>Clk	Boolean	AXI-Stream RX interface, user signals from core. 0 indicates that a bad packet has been received. 1 indicates that a good packet has been received. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1RxTUser	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane2RxTUser	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane3RxTUser	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane0RxTKeep	Out	UserQsfp<0,1>Clk	Unsigned 8-bit	AXI-Stream RX interface, data control from core to upper layer. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1RxTKeep	Out	UserQsfp<0,1>Clk	Unsigned 8-bit	
uQsfp<0,1>Lane2RxTKeep	Out	UserQsfp<0,1>Clk	Unsigned 8-bit	
uQsfp<0,1>Lane3RxTKeep	Out	UserQsfp<0,1>Clk	Unsigned 8-bit	

Port	Direction	Clock Domain	Type	Description
uQsfp<0,1>Lane0RxTReady	In	UserQsfp<0,1>Clk	Boolean	AXI-Stream RX interface, acknowledge signals from core to indicate the start of a data transfer. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1RxTReady	In	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane2RxTReady	In	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane3RxTReady	In	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane0PhyStatus	Out	UserQsfp<0,1>Clk	Unsigned 8-bit	PHY status vectors. Bits 7 down to 1 are unused. Bit-0 indicates the current status of the link (0: link down, 1: link up). Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1PhyStatus	Out	UserQsfp<0,1>Clk	Unsigned 8-bit	
uQsfp<0,1>Lane2PhyStatus	Out	UserQsfp<0,1>Clk	Unsigned 8-bit	
uQsfp<0,1>Lane3PhyStatus	Out	UserQsfp<0,1>Clk	Unsigned 8-bit	
uQsfp<0,1>Lane0MacStatus	Out	UserQsfp<0,1>Clk	Unsigned 9-bit	MAC status vectors. Bit-0: CRC Error bit-1: Fragment Error bit-2: TX FIFO Overflow bit-3: TX FIFO Underflow bit-4: RX FIFO Overflow bit-5: RX FIFO Underflow bit-6: Pause Frame RX bit-7: Local Fault bit-8: Remote Fault
uQsfp<0,1>Lane1MacStatus	Out	UserQsfp<0,1>Clk	Unsigned 9-bit	
uQsfp<0,1>Lane2MacStatus	Out	UserQsfp<0,1>Clk	Unsigned 9-bit	
uQsfp<0,1>Lane3MacStatus	Out	UserQsfp<0,1>Clk	Unsigned 9-bit	

Port	Direction	Clock Domain	Type	Description
				Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane0Activity	Out	UserQsfp<0,1>Clk	Boolean	Indicates activity in the link. These signals go through a pulse stretcher to allow visualization, if required. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1Activity	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane2Activity	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane3Activity	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane0LinkUp	Out	UserQsfp<0,1>Clk	Boolean	Indicates the current status of the link. These signals are the same as Bit 0 from the corresponding uQsfp<0,1>Lane*PhyStatus. Lanes 1, 2, and 3 are only available in 4 x 10 GbE CLIP.
uQsfp<0,1>Lane1LinkUp	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane2LinkUp	Out	UserQsfp<0,1>Clk	Boolean	
uQsfp<0,1>Lane3LinkUp	Out	UserQsfp<0,1>Clk	Boolean	

Using the Ettus USRP X410 with the LabVIEW FPGA Module

Complete the following steps to route a signal into or out of the LabVIEW FPGA Module using the TRIG IN/OUT port on the Ettus USRP X410 device. This allows you to provide an external trigger signal for synchronous triggering of your application.

1. Verify that you have downloaded compatible versions of NI-USRP and LabVIEW FPGA. Check your product Readme for details.
2. Optionally, change the behavior of the FPGA image.
 - a. Open the Reference FPGA Project from **<LabVIEW directory> » instr.lib » niUSRP » Reference FPGA**.
 - b. In the Reference FPGA Project, the Trig Out Routing VI shows the default behavior, which is to route PPS IN to TRIG OUT. A TRIG IN object can also be added from the Project Explorer window. To change the default behavior, TRIG IN and TRIG OUT can be routed to VIs that you add to the project from the FPGA I/O palette
 - c. Compile the FPGA.
3. Specify the desired interaction between the physical Trig Input/Output port and objects in LabVIEW FPGA.
 - a. Adjust the Trig I/O Mode property within the niUSRP Property Node VI. The options for the property are Input, Output, and Off.

Table 7. niUSRP Property Node VI Properties

Property	Description
Off	By default, the property is set to Off. In this state, the TRIG IN/OUT port is not in use.
Input	When the property is set to Input, the signal that is going into the TRIG IN/OUT port on the device is routed to the TRIG IN object in LabVIEW FPGA.
Output	When the property is set to Output, the signal from the TRIG OUT object in LabVIEW FPGA is routed to the TRIG IN/OUT port on the device.

- b. For applications where the Trig I/O Mode property in the niUSRP Property Node VI is set to Input or Output, ensure that an SMA (m)-to-SMA (m) cable connects the appropriate hardware to the TRIG IN/OUT port on your device.