LogiCORE IP Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper v2.2

DS835 October 19, 2011

Product Specification

Introduction

The LogiCORE™ IP Virtex®-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper is comprised of the Embedded Tri-Mode Ethernet MAC primitive with additional logic to simplify and update the user interface. It is available in Virtex-6 LXT, SXT, HXT, and CXT FPGAs using the Xilinx® CORE Generator™ software.

Features

- Sets the Ethernet MAC attributes based on user options
- Provides user-configurable Ethernet MAC physical interfaces
  - Supports Reduced Gigabit Media Independent Interface (RGMII) v2.0 [Ref 2], Serial Gigabit Media Independent Interface (SGMII), and 1000BASE-X PCS/PMA interfaces, as well as Gigabit Media Independent Interface (GMII)/Media Independent Interface (MII) at 2.5V only
  - Instantiates clock buffers, MMCMs, GTX serial transceivers, and logic as required for the selected physical interfaces
- Generates VHDL or Verilog
- Configured and monitored through an optional AXI4-Lite interface
- Optional MDIO interface to managed objects in PHY layers (MII Management)
- Optional Frame Filter with selectable number of address table entries
- Supports VLAN frames, jumbo frames and allows a configurable interframe gap
- Configurable in-band Frame Check Sequence (FCS) field passing on both transmit and receive paths
- Optional built-in statistics counters
- Supports AXI4-Stream on RX and TX datapaths
- No charge IP core available under the End User License Agreement

LogiCORE IP Facts Table				
Core Specifics				
Supported Device Family(1)	Virtex-6 LXT, SXT, HXT, and CXT			
Supported User Interfaces	AXI4-Lite, AXI4-Stream			
Performance	10 Mb/s, 100 Mb/s, 1 Gb/s(2),2 Gb/s(2), 2.5 Gb/s(2)			
Resources(3)				
	LUTs	**FFs**	**Slices**	**BUFG**
	200-1200	200-1500	100-800	2-4
Provided with Core				
Documentation	Product Specification User Guide			
Design Files	NGC netlist			
Example Design	VHDL and Verilog			
Test Bench	Demonstration Test Bench			
Constraints File	UCF			
Simulation Model	Verilog SecureIP model(4)			
Tested Design Tools				
Design Entry Tools	ISE software v13.3			
Simulation(6)	Mentor Graphics ModelSim Cadence Incisive Enterprise Simulator (IES) Synopsys VCS and VCS MX (5)			
Synthesis Tools	XST 13.3			
Support				
Provided by Xilinx, Inc.				

1. For a complete list of supported devices, see the release notes for this core.
2. Performance is subject to device support. See Performance.
3. See Table 37 to Table 39; precise number depends on user configuration and family.
4. Requires a Verilog LRM-IEEE 1364-2005 encryption-compliant simulator. For VHDL simulation, a mixed HDL license is required.
5. Scripts provided for listed simulators only.
6. For the supported version of the tools, see the ISE Design Suite 13: Release Notes Guide

Applications

Typical applications for the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper include the following:

- Ethernet 1000BASE-X Port
- Ethernet Tri-Speed BASE-T Port

Ethernet 1000BASE-X Port

Figure 1 illustrates a typical application for an Ethernet MAC. The PHY side of the MAC is connected to a GTX serial transceiver, which in turn is connected to an external off-the-shelf GBIC or SFP optical transceiver. The 1000BASE-X PCS/PMA logic can be optionally provided by the Ethernet MAC, as displayed. 1000BASE-X functionality is demonstrated in the HDL examples provided with the example design.

The user side of the core is shown connected to the 10 Mb/s, 100 Mb/s, 1 Gb/s Ethernet FIFO, delivered with the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper to complete a single Ethernet port. This port is shown connected to a Switch or Routing matrix, which can contain several ports.

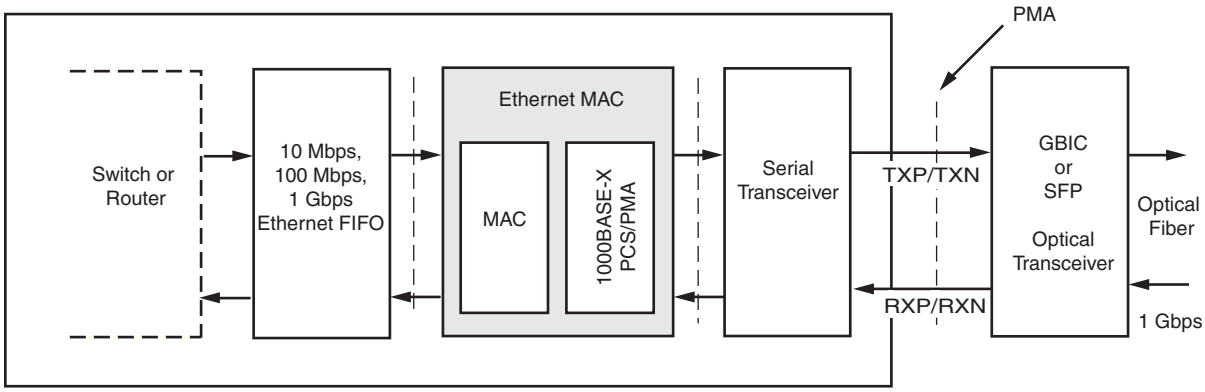

Figure 1: **Typical MAC 1000BASE-X Application**

Ethernet Tri-Speed BASE-T Port

Figure 2 illustrates a typical application for the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper (10/100/1000 Mb/s) core. The PHY side of the core is implementing an external GMII/MII by connecting it to IOBs; the external GMII/MII is connected to an off-the-shelf Ethernet PHY device, which performs the BASE-T standard at 1 Gb/s, 100 Mb/s, and 10 Mb/s speeds. Alternatively, the external GMII/MII can be replaced with an RGMII (as shown) or as an SGMII (which requires the use of a GTX serial transceiver). GMII, RGMII, and SGMII functionality are demonstrated in the HDL examples provided with the example design.

The user side of the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper is shown connected to the 10 Mb/s, 100 Mb/s, 1 Gb/s Ethernet FIFO (delivered with the example design) to complete a single Ethernet port. This port is shown connected to a Switch or Routing matrix, which can contain several ports.

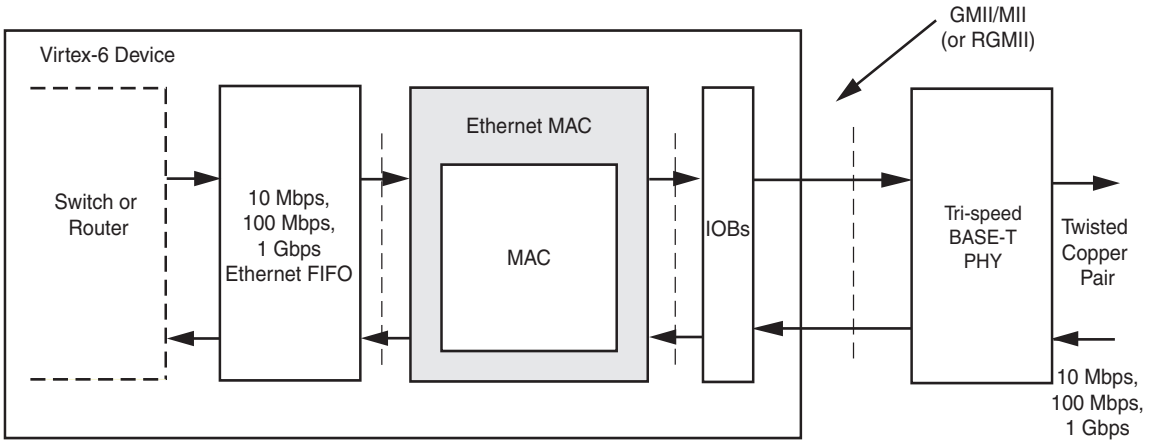

Figure 2: **Typical BASE-T Application for the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper Core**

Ethernet Architecture Overview

The MAC sublayer provided by this core is part of the Ethernet architecture displayed in Figure 3. The portion of the architecture, from the MAC to the right, is defined in *IEEE 802.3*. This figure also illustrates where the supported interfaces fit into the architecture.

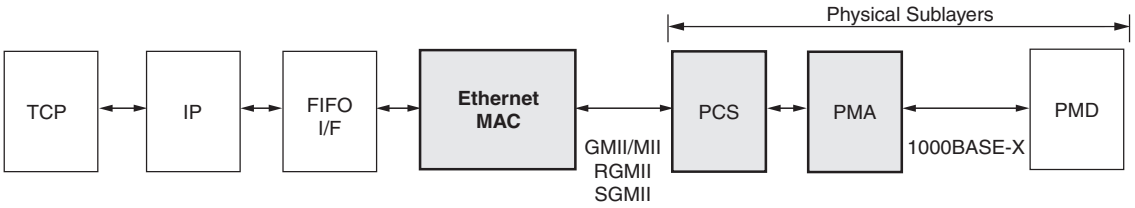

Figure 3: **Typical Ethernet Architecture**

MAC

The Ethernet Medium Access Controller (MAC) is defined in [Ref 1] clauses 2, 3, and 4. A MAC is responsible for the Ethernet framing protocols and error detection of the frames. The MAC is independent of, and can be connected to, any type of physical layer.

GMII / MII

The Media Independent Interface (MII), defined in *IEEE 802.3* clause 22, is a parallel interface that connects a 10-Mb/s and/or 100-Mb/s capable MAC to the physical sublayers. The Gigabit Media Independent Interface (GMII), defined in *IEEE 802.3* clause 35, is an extension of the MII used to connect a 1-Gb/s capable MAC to the physical sublayers. MII can be considered a subset of GMII, and as a result, GMII/MII can carry Ethernet traffic at 10 Mb/s, 100 Mb/s, and 1 Gb/s. GMII/MII is supported at 2.5V only. See [Ref 3].

RGMII

The Reduced Gigabit Media Independent Interface (RGMII) is an alternative to the GMII. RGMII achieves a 50-percent reduction in the pin count, compared with GMII, and for this reason is preferred over GMII by PCB designers. This is achieved with the use of double-data-rate (DDR) flip-flops.

SGMII

The Serial-GMII (SGMII) interface is an alternative to GMII/MII. SGMII converts the parallel interface of the GMII/MII into a serial format using a GTX serial transceiver, radically reducing the I/O count. For this reason, it is often the preferred interface of PCB designers. SGMII can carry Ethernet traffic at 10 Mb/s, 100 Mb/s, and 1 Gb/s.

PCS, PMA, and PMD

The combination of the Physical Coding Sublayer (PCS), the Physical Medium Attachment (PMA), and the Physical Medium Dependent (PMD) sublayer comprise the physical layers of the Ethernet protocol.

Two main physical standards are specified for Ethernet:

- BASE-T, a copper standard using twisted pair cabling systems
- BASE-X, usually a fiber optical physical standard using short and long wavelength laser

BASE-T devices, supporting 10 Mb/s, 100 Mb/s, and 1 Gb/s Ethernet speeds, are readily available as off-the-shelf parts. As illustrated in Figure 3 and Figure 2, these can be connected using GMII/MII, RGMII, or SGMII to provide a tri-speed Ethernet port.

The Ethernet MAC has built-in 1000BASE-X PCS/PMA functionality and can be connected to a GTX serial transceiver to provide a 1 Gb/s fiber optic port, as illustrated in Figure 1.

Block Overview

Figure 4 identifies the major functional blocks of the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper. Descriptions of the functional blocks and interfaces are provided in the subsequent sections.

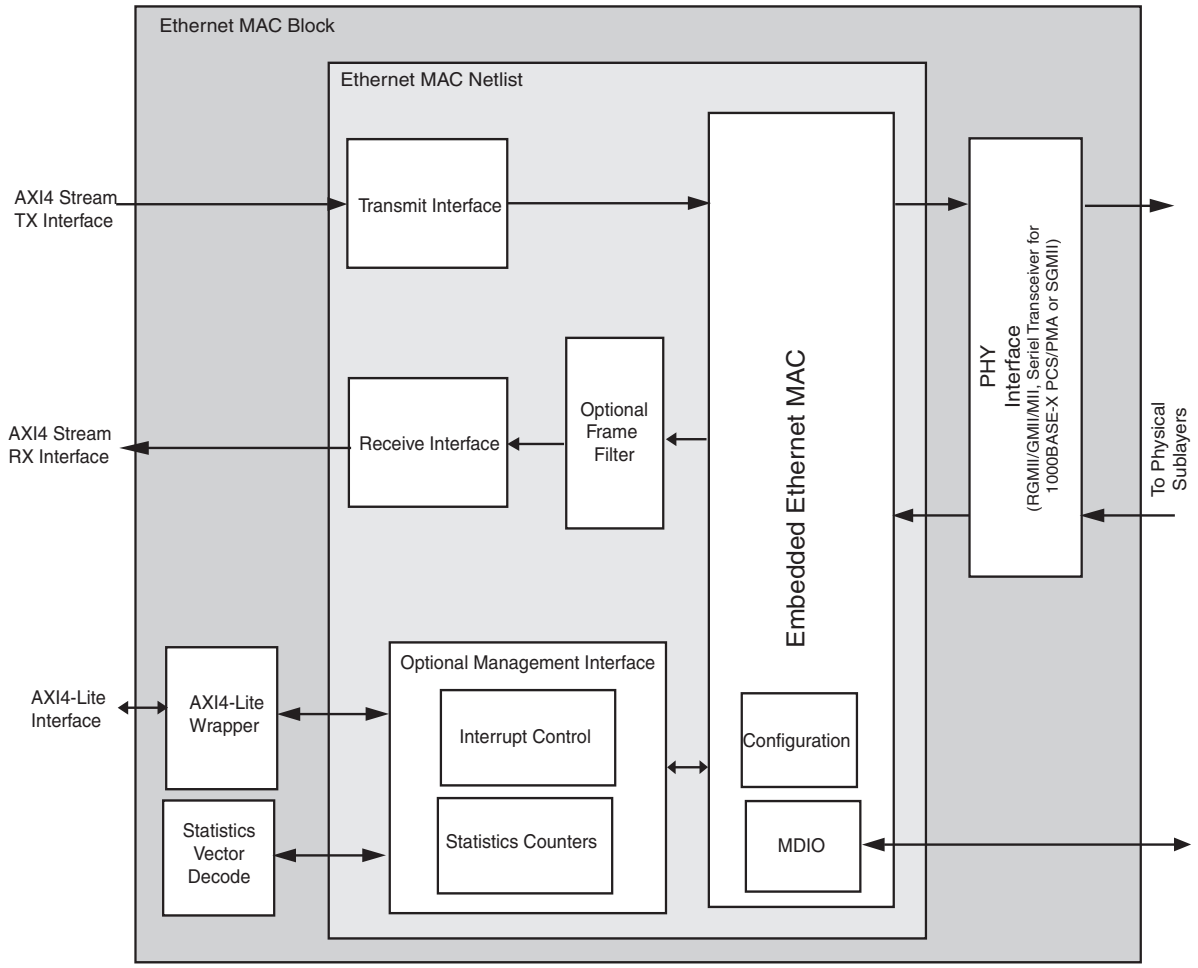

Figure 4: **Virtex-6 Embedded TEMAC Functional Block Diagram**

Ethernet Mac Block

The Ethernet MAC block is provided as part of the HDL example design and includes the basic blocks required to use the Ethernet MAC netlist. The Ethernet MAC Block should be instantiated in all designs that use the core.

AXI4-Lite Wrapper

The AXI4-Lite Wrapper allows the Ethernet MAC netlist to be connected to an AXI4-Lite Interface and drives the Ethernet MAC netlist through a processor independent IPIF.

Statistics Vector Decode

The Statistics Vector Decode interprets the RX and TX statistics vectors supplied by the Ethernet MAC netlist on a per-frame basis and generates the statistics counter increment controls. This code is provided as editable HDL to enable specific statistics counter requirements to be met.

PHY Interface

The PHY Interface provides the required logic to interface to the PHY device.

When using either RGMII or GMII/MII, this comprises an interface-specific block which instantiates the IOB logic and clock resources required.

The core can be optionally generated with 1000BASE-X PCS/PMA logic providing support for 1000BASE-X and SGMII. Both these standards require the use of a GTX serial transceiver. This wrapper provides reset logic and correctly sets attributes for the required standard.

Transmit Interface

The transmit interface takes data from the AXI4-Stream TX interface and converts it to the client format expected by the Embedded Ethernet MAC.

Receive Interface

The receive interface takes the data from the Embedded Ethernet MAC client RX interface and converts it to the AXI4-Stream RX format expected by the user.

Management Interface

The optional Management Interface converts between the processor-independent interface and the Host interface expected by the core. It is also responsible for mapping between the new fully address-mapped register set and the Embedded Ethernet MAC registers. It is used for the configuration and monitoring of the Ethernet MAC, and for access to the MDIO Interface. It is supplied with a wrapper to interface to the industry-standard AXI4-Lite. This interface is optional. If it is not present, the device attributes are taken from those specified at generation.

MDIO Interface

The optional MDIO interface can be written to, and read from, using the Management Interface. The MDIO is used to monitor and configure both internal and external PHY devices. When no management interface is used, the MDIO interface operates as a slave, allowing configuration of the internal PHY. The MDIO Interface is defined in *IEEE 802.3* clause 22.

Frame Filter

The Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper can be implemented with an optional Frame Filter. If the Frame Filter is enabled, the device does not pass frames that do not match against either a known address or one of the configurable frame filters.

Statistics Counters

The Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper can be implemented with optional Statistics Counters. See [Ref 3] for more details.

Embedded Ethernet MAC

The Embedded Ethernet MAC has a large number of attributes and I/O. These are tied off as appropriate based on the core configuration selected by the user. The Embedded Ethernet MAC is described in more detail in [Ref 3].

Interface Descriptions

All ports of the netlist are internal connections in the Field Programmable Gate Array (FPGA) logic. An example HDL design, provided in both VHDL and Verilog, is delivered with each core. The example design connects the core to a FIFO-based loopback example design and adds IOB flip-flops to the external signals of the GMII/MII (or RGMII) or a GTX transceiver for SGMII and 1000BASE-X.

All clock management logic is placed in this example design, allowing you more flexibility in implementation (for example, in designs using multiple cores). For information about the example design, see [Ref 3].

Transmitter Interface

Signal Definition

Table 1 defines the AXI4-Stream transmit signals of the core, which are used to transmit data from the user to the core. Table 2 defines transmit sideband signals.

Table 1: **Transmit Interface AXI4-Stream Signal Pins**

Signal	Direction	Clock Domain	Description
tx_axis_mac_tdata[7:0] or [15:0](2)	Input	tx_mac_aclk	Frame data to be transmitted.
tx_axis_mac_tkeep[1:0]	Input	tx_mac_aclk	Control signal for tx_axis_mac_tdata port. Indicates which bytes of tdata[15:0] are valid (16-bit AXI4- Stream Interface only).
tx_axis_mac_tvalid	Input	tx_mac_aclk	Control signal for tx_axis_mac_tdata port. Indicates the data is valid.
tx_axis_mac_tlast	Input	tx_mac_aclk	Control signal for tx_axis_mac_tdata port. Indicates the final transfer in a frame.
tx_axis_mac_tuser	Input	tx_mac_aclk	Control signal for tx_axis_mac_tdata port. Indicates an error condition, such as FIFO underrun, in the frame allowing the MAC to send an error to the PHY.
tx_axis_mac_tready	Output	tx_mac_aclk	Handshaking signal. Asserted when the current data on tx_axis_mac_tdata has been accepted and tx_axis_mac_tvalid is high. At 10/100Mb/s this is used to meter the data into the core at the correct rate.

Notes:

1. All signals are active high
2. The 16-bit option is only available when using the 1000BASE-X PCS/PMA physical interface and the 2 or 2.5 Gb/s overclocking option.

Table 2: **Transmit Interface Sideband Signal Pins**

Signal	Direction	Clock Domain	Description
tx_ifg_delay[7:0]	Input	tx_mac_aclk	Control signal for configurable interframe gap
tx_collision	Output	tx_mac_aclk	Asserted by the MAC netlist to signal a collision on the medium and that any transmission in progress should be aborted. Always 0 when the MAC netlist is in full-duplex mode.
tx_retransmit	Output	tx_mac_aclk	When asserted at the same time as the tx_collision signal, this signals to the client that the aborted frame should be resupplied to the MAC netlist for retransmission. Always 0 when the MAC netlist is in full-duplex mode.

Table 2: **Transmit Interface Sideband Signal Pins *(Cont'd)***

Signal	Direction	Clock Domain	Description
tx_statisitics_vector[31:0]	Output	tx_mac_aclk	A statistics vector that gives information on the last frame transmitted.
tx_statistics_valid	Output	tx_mac_aclk	Asserted at end of frame transmission, indicating that the tx_statistics_vector is valid.

Notes:
1. All signals are active high.

Transmitter AXI4-Stream Interface Timing

Figure 5 displays a typical frame transmission at the user interface. All signals are synchronous to the `tx_mac_aclk` clock. See [Ref 3] for further information.

To transmit a frame, the user asserts `tx_axis_mac_tvalid` and puts the first byte of frame data on the `tx_axis_mac_tdata` bus. The user then waits until the core asserts `tx_axis_mac_tready` before providing the next byte of data. The user must be capable of providing new data on the cycle after `tx_axis_mac_tready` is asserted at all times; there is no way for the user to throttle the data. On the final byte of the frame, `tx_axis_mac_tlast` is asserted.

At 1 Gb/s, data can be taken every 8 ns; at 100 Mb/s, data is taken, on average, every 80 ns; at 10 Mb/s, data is taken, on average, every 800 ns. In all cases `tx_axis_mac_tready` qualifies when data is taken by the MAC.

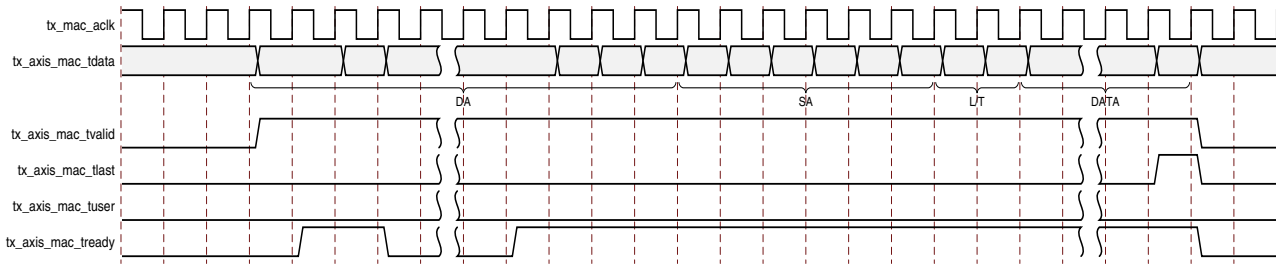

Figure 5: **Normal Frame Transmission across AXI4-Stream Interface**

Receiver Interface

Signal Definition

Table 3 describes the receive AXI4-Stream signals used by the core to transfer data to the user. Table 4 describes the related sideband interface signals.

Table 3: **Receive Interface AXI4-Stream Signal Pins**

Signal	Direction	Clock Domain	Description
rx_axis_mac_tdata[7:0] or [15:0](2)	Output	rx_mac_aclk	Frame data received is supplied on this port.
rx_axis_mac_tkeep[1:0]	Output	rx_mac_aclk	Control signal for rx_axis_mac_tdata port. Indicates which bytes of tdata[15:0] are valid (16-bit AXI4- Stream Interface only).
rx_axis_mac_tvalid	Output	rx_mac_aclk	Control signal for the rx_axis_mac_tdata port. Indicates the data is valid.
rx_axis_mac_tlast	Input	rx_mac_aclk	Control signal for the rx_axis_mac_tdata port. Indicates the final byte in the frame.

Table 3: **Receive Interface AXI4-Stream Signal Pins *(Cont'd)***

Signal	Direction	Clock Domain	Description
rx_axis_mac_tuser	Output	rx_mac_aclk	Control signal for rx_axis_mac_tdata. Asserted at end of frame reception to indicate that the frame had an error.
rx_axis_filter_tuser[x:0]	Output	rx_mac_aclk	Per Frame filter tuser output. Can be used to send only data passed by a specific Frame filter. See [Ref 3].

Notes:

1. All signals are active high.
2. The 16-bit option is only available when using the 1000BASE-X PCS/PMA physical interface and the 2 or 2.5 Gb/s overclocking option.

Table 4: **Receive Interface Sideband Signal Pins**

Signal	Direction	Clock Domain	Description
rx_statistics_vector[27:0]	Output	rx_mac_aclk	Provides information about the last frame received.
rx_statistics_valid	Output	rx_mac_aclk	Asserted at end of frame reception, indicating that the rx_statistics_vector is valid.

Note: All signals are active high.

Receiver AXI4-Stream Interface Timing

Figure 6 displays the reception of a good frame at the user interface. All signals are synchronous to the `rx_mac_aclk` clock.

When receiving a frame, the core asserts `rx_axis_mac_tvalid` for each valid byte of frame data. On the final byte of the frame, `rx_axis_mac_tlast` is asserted as well as `rx_axis_mac_tvalid`. `rx_axis_mac_tuser` can also be asserted for the final byte of the frame to indicate that the frame included an error.

Note: The core does not have any way for the user to throttle the data; it is assumed that the user is always able to take data when presented by the MAC.

At 1 Gb/s, data can be presented every 8 ns; at 100 Mb/s, data can be presented, on average, every 80 ns; at 10 Mb/s, data can be presented, on average, every 800 ns.

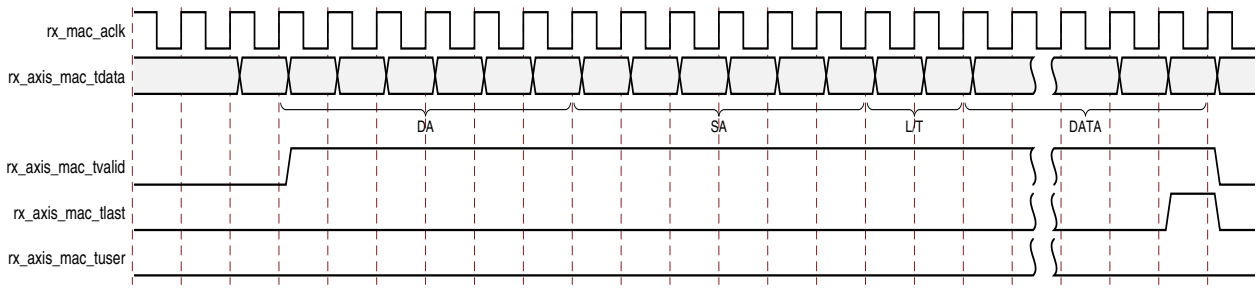

Figure 6: **Normal Frame Reception at AXI4-Stream Interface**

Flow Control User Side Interface Signal Definition

Table 5 describes the signals used to request a flow-control action from the transmit engine. Valid flow control frames received by the MAC are automatically handled (if the MAC is configured to do so). The pause value in the received frame is used to inhibit the transmitter operation for the time defined in [Ref 1]. The frame is then passed to the client with `rx_axis_mac_tuser` asserted to indicate to the client that it should be dropped.

Table 5: **Flow Control Interface Signal Pinout**

Signal	Direction	Description
pause_req	Input	Pause request: Upon request the MAC transmits a pause frame upon the completion of the current data packet.
pause_val[15:0]	Input	Pause value: inserted into the parameter field of the transmitted pause frame.

Note: All signals are active high.

AXI4-Lite Signal Definition

Table 6 describes the optional signals used by the user to access the MAC netlist, including configuration, status and MDIO access.

Table 6: **Optional AXI4-Lite Signal Pinout**

Signal	Direction	Clock Domain	Description
s_axi_aclk	Input	N/A	Clock for AXI4-Lite
s_axi_resetn	Input	s_axi_aclk	Local reset for the clock domain
s_axi_awaddr[31:0]	Input	s_axi_aclk	Write Address
s_axi_awvalid	Input	s_axi_aclk	Write Address Valid
s_axi_awready	Output	s_axi_aclk	Write Address ready
s_axi_wdata[31:0]	Input	s_axi_aclk	Write Data
s_axi_wvalid	Input	s_axi_aclk	Write Data valid
s_axi_wready	Output	s_axi_aclk	Write Data ready
s_axi_bresp[1:0]	Output	s_axi_aclk	Write Response
s_axi_bvalid	Output	s_axi_aclk	Write Response valid
s_axi_bready	Input	s_axi_aclk	Write Response ready
s_axi_araddr[31:0]	Input	s_axi_aclk	Read Address
s_axi_arvalid	Input	s_axi_aclk	Read Address valid
s_axi_arready	Output	s_axi_aclk	Read Address ready
s_axi_rdata[31:0]	Output	s_axi_aclk	Read Data
s_axi_rresp[1:0]	Output	s_axi_aclk	Read Response
s_axi_rvalid	Output	s_axi_aclk	Read Data/Response Valid
s_axi_rready	Input	s_axi_aclk	Read Data/Response ready

Clock, Speed Indication, and Reset Signal Definition

Table 7 describes the reset signals, the clock signals that are input to the core, and the outputs that can be used to select between the three operating speeds. The clock signals are generated in the top-level wrapper provided with the core.

Table 7: **Clock and Speed Indication Signals**

Signal	Direction	Description
glbl_rstn	Input	Active low asynchronous reset for entire core.
rx_axi_rstn	Input	Active low RX domain reset
tx_axi_rstn	Input	Active low TX domain reset
rx_reset	Output	Active high RX software reset from MAC netlist
tx_reset	Output	Active high TX software reset from MAC netlist
gtx_clk	Input	Global 125MHz clock (or a higher frequency when using the overclocked 1000BASE-X physical interface)
tx_mac_aclk	Input	Clock for the transmission of data on the physical interface. 125 MHz at 1 Gb/s, 25 MHz at 100 Mb/s, and 2.5 MHz at 10 Mb/s. This clock should be used to clock the physical interface transmit circuitry and the TX AXI4-Stream transmit circuitry. This clock only exists in GMII or MII.
rx_mac_aclk	Input	Clock for the reception of data on the physical interface. 125 MHz at 1 Gb/s, 25 MHz at 100 Mb/s, and 2.5 MHz at 10 Mb/s. This clock should be used to clock the physical interface receive circuitry and the RX AXI4-Stream receive circuitry.
speedis100	Output	Output asserted when the core is operating at 100 Mb/s. It is derived from a configuration register.
speedis10100	Output	This output is asserted when the core is operating at either 10 Mb/s or 100 Mb/s. It is derived from a configuration register.

Physical Interface Signal Definition

Table 8 describes the MDIO (MII Management) interface signals of the core, which are typically connected to the MDIO port of a PHY device, either off-chip or an SoC-integrated core. The MDIO format is defined in [Ref 1] clause 22.

Table 8: **MDIO Interface Signal Pinout**

Signal	Direction	Description
mdc	Output	MDIO Management Clock: derived from s_axi_aclk on the basis of supplied configuration data when the optional Management Interface is used.
mdc_in	Input	MDIO Management Clock input. Present when either SGMII or 1000BASE-X are used and the optional management interface is not selected.
mdio_i	Input	Input data signal for communication with PHY configuration and status. Tie high if unused.
mdio_o	Output	Output data signal for communication with PHY configuration and status.
mdio_t	Output	3-state control for MDIO signals; '0' signals that the value on MDIO_OUT should be asserted onto the MDIO bus.

Table 9 through Table 11 describe the three parallel interface standards and two serial standards supported, RGMII, GMII and MII, which are typically attached to an off-chip PHY module and SGMII and 1000BASE-X The RGMII is defined in [Ref 2], the GMII is defined in [Ref 1] clause 35, and MII is defined in [Ref 1] clause 22.

Table 9: **Optional GMII Interface Signal Pinout**

Signal	Direction	Clock Domain	Description
gmii_txd[7:0]	Output	tx_mac_aclk	Transmit data to PHY
gmii_tx_en	Output	tx_mac_aclk	Data Enable control signal to PHY
gmii_tx_er	Output	tx_mac_aclk	Error control signal to PHY
mii_tx_clk	Input		Clock from PHY (used for 10/100)
gmii_col	Input	N/A	Control signal from PHY
gmii_crs	Input	N/A	Control signal from PHY
gmii_rxd[7:0]	Input	gmii_rx_clk	Received data from PHY
gmii_rx_dv	Input	gmii_rx_clk	Data Valid control signal from PHY
gmii_rx_er	Input	gmii_rx_clk	Error control signal from PHY
gmii_rx_clk	Input		Clock from PHY

Table 10: **Optional MII Interface Signal Pinout**

Signal	Direction	Clock Domain	Description
mii_tx_clk	Input		Clock from PHY
mii_txd[3:0]	Output	mii_tx_clk	Transmit data to PHY
mii_tx_en	Output	mii_tx_clk	Data Enable control signal to PHY
mii_tx_er	Output	mii_tx_clk	Error control signal to PHY
mii_col	Input	N/A	Control signal from PHY
mii_crs	Input	N/A	Control signal from PHY
mii_rxd[3:0]	Input	rx_mac_aclk	Received data from PHY
mii_rx_dv	Input	rx_mac_aclk	Data Valid control signal from PHY
mii_rx_er	Input	rx_mac_aclk	Error control signal from PHY
mii_rx_clk	Input		Clock from PHY

Table 11: **Optional RGMII Interface Signal Pinout**

Signal	Direction	Clock Domain	Description
rgmii_txd[3:0]	Output	tx_mac_aclk	Transmit data to PHY
rgmii_tx_ctl	Output	tx_mac_aclk	control signal to PHY
rgmii_txc	Output		Clock to PHY
rgmii_rxd[3:0]	Input	rgmii_rxc	Received data from PHY
rgmii_rx_ctl	Input	rgmii_rxc	Control signal from PHY
rgmii_rxc	Input		Clock from PHY

Table 12: **Optional SGMII or 1000BASE-X Interface Signal Pinout**

Signal	Direction	Clock Domain	Description
txp	Output		Transmit data from GTX transceiver
txn	Output		Transmit data from GTX transceiver
rxp	Output		Receive data to GTX transceiver
rxn	Input		Receive data to GTX transceiver

Control and Status Registers

When the core is generated with a management interface, all control and status registers are memory mapped. If no management interface is used, the key core attributes are set based on the GUI settings. After power up or reset, the user can reconfigure the core parameters from their defaults, such as flow control support. Configuration changes can be made at any time. Both the receiver and transmitter logic only sample configuration changes at the start of frame transmission/reception. The exceptions to this are the configurable resets which take effect immediately.

Configuration of the core is performed through a register bank accessed through the AXI4-Lite interface. The configuration registers available in the core are detailed in Table 13.

Table 13: **Core Registers**

Address	Description
0x000-0x1FC	Reserved
0x200	Received Bytes Counter word 0
0x204	Received Bytes Counter word 1 (if 64 bit width)
0x208	Transmitted Bytes Counter word 0
0x20C	Transmitted Bytes Counter word 1 (if 64 bit width)
0x210	Undersize Frames Counter word 0
0x214	Undersize Frames Counter word 1 (if 64 bit width)
0x218	Fragment Frames Counter word 0
0x21C	Fragment Frames Counter word 1 (if 64 bit width)
0x220	RX 64 Byte Frames Counter word 0
0x224	RX 64 Byte Frames Counter word 1 (if 64 bit width)
0x228	RX 65-127 Byte Frames Counter word 0
0x22C	RX 65-127 Byte Frames Counter word 1 (if 64 bit width)
0x230	RX 128-255 Byte Frames Counter word 0
0x234	RX 128-255 Byte Frames Counter word 1 (if 64 bit width)
0x238	RX 256-511 Byte Frames Counter word 0
0x23C	RX 256-511 Byte Frames Counter word 1 (if 64 bit width)
0x240	RX 512-1023 byte Frames Counter word 0
0x244	RX 512-1023 Byte Frames Counter word 1 (if 64 bit width)
0x248	RX 1024-Max Frames Size Byte Frames Counter word 0
0x24C	RX 1024-Max Frames Size Byte Frames Counter word 1 (if 64 bit width)
0x250	RX Oversize Frames Counter word 0
0x254	RX Oversize Frames Counter word 1 (if 64 bit width)
0x258	TX 64 Byte Frames Counter word 0

Table 13: **Core Registers *(Cont'd)***

Address	Description
0x25C	TX 64 Byte Frames Counter word 1 (if 64 bit width)
0x260	TX 65-127 Byte Frames Counter word 0
0x264	TX 65-127 Byte Frames Counter word 1 (if 64 bit width)
0x268	TX 128-255 Byte Frames Counter word 0
0x26C	TX 128-255 Byte Frames Counter word 1 (if 64 bit width)
0x270	TX 256-511 Byte Frames Counter word 0
0x274	TX 256-511 Byte Frames Counter word 1 (if 64 bit width)
0x278	TX 512-1023 byte Frames Counter word 0
0x27C	TX 512-1023 Byte Frames Counter word 1 (if 64 bit width)
0x280	TX 1024-Max Frames Size Byte Frames Counter word 0
0x284	TX 1024-Max Frames Size Byte Frames Counter word 1 (if 64 bit width)
0x288	TX Oversize Frames Counter word 0
0x28C	TX Oversize Frames Counter word 1 (if 64 bit width)
0x290	RX Good Frames Counter word 0
0x294	RX Good Frames Counter word 1 (if 64 bit width)
0x298	RX Frame Check Sequence Errors Counter word 0
0x29C	RX Frame Check Sequence Errors Counter word 1 (if 64 bit width)
0x2A0	RX Good Broadcast Frames Counter word 0
0x2A4	RX Good Broadcast Frames Counter word 1 (if 64 bit width)
0x2A8	RX Good Multicast Frames Counter word 0
0x2AC	RX Good Multicast Frames Counter word 1 (if 64 bit width)
0x2B0	RX Good Control Frames Counter word 0
0x2B4	RX Good Control Frames Counter word 1 (if 64 bit width)
0x2B8	RX Length/Type Out of Range Errors Counter word 0
0x2BC	RX Length/Type Out of Range Errors Counter word 1 (if 64 bit width)
0x2C0	RX Good VLAN Tagged Frames Counter word 0
0x2C4	RX Good VLAN Tagged Frames Counter word 1 (if 64 bit width)
0x2C8	RX Good Pause Frames Counter word 0
0x2CC	RX Good Pause Frames Counter word 1 (if 64 bit width)
0x2D0	RX Bad Opcode Frames Counter word 0
0x2D4	RX Bad Opcode Frames Counter word 1 (if 64 bit width)
0x2D8	TX Good Frames Counter word 0
0x2DC	TX Good Frames Counter word 1 (if 64 bit width)
0x2E0	TX Good Broadcast Frames Counter word 0
0x2E4	TX Good Broadcast Frames Counter word 1 (if 64 bit width)
0x2E8	TX Good Multicast Frames Counter word 0
0x2EC	TX Good Multicast Frames Counter word 1 (if 64 bit width)
0x2F0	TX Underrun Errors Counter word 0

Table 13: **Core Registers *(Cont'd)***

Address	Description
0x2F4	TX Underrun Errors Counter word 1 (if 64 bit width)
0x2F8	TX Good Control Frames Counter word 0
0x2FC	TX Good Control Frames Counter word 1 (if 64 bit width)
0x300	TX Good VLAN Frames Counter word 0
0x304	TX Good VLAN Frames Counter word 1 (if 64 bit width)
0x308	TX Good Pause Frames Counter word 0
0x30C	TX Good Pause Frames Counter word 1 (if 64 bit width)
0x310	TX Single Collision Frames Counter word 0
0x314	TX Single Collision Frames Counter word 1 (if 64 bit width)
0x318	TX Multiple Collision Frames Counter word 0
0x31C	TX Multiple Collision Frames Counter word 1 (if 64 bit width)
0x320	TX Deferred Frames Counter word 0
0x324	TX Deferred Frames Counter word 1 (if 64 bit width)
0x328	TX Late Collision Counter word 0
0x32C	TX Late Collision Counter word 1 (if 64 bit width)
0x330	TX Excess Collision Counter word 0
0x334	TX Excess Collision Counter word 1 (if 64 bit width)
0x338	TX Excess Deferral Counter word 0
0x33C	TX Excess Deferral Counter word 1 (if 64 bit width)
0x340	TX Alignment Errors Counter word 0
0x344	TX Alignment Errors Counter word 1 (if 64 bit width)
0x348-0x364	User Defined Statistics Counters (if present)
0x368-0x3FC	Reserved
0x400	Receiver Configuration word 0
0x404	Receiver Configuration word 1
0x408	Transmitter configuration
0x40C	Flow Control Configuration
0x410	Ethernet MAC Mode configuration
0x414-0x41C	Reserved
0x420	RGMII/SGMII Configuration
0x424-0x4F4	Reserved
0x4F8	ID Register
0x4FC	Ability Register
0x500	MDIO Setup
0x504	MDIO Control
0x508	MDIO Write Data
0x50C	MDIO Read Data
0x510-0x5FC	Reserved

Table 13: **Core Registers *(Cont'd)***

Address	Description
0x600	Interrupt Status Register
0x604-0x60C	Reserved
0x610	Interrupt Pending Register
0x614-0x61C	Reserved
0x620	Interrupt Enable Register
0x624-0x62C	Reserved
0x630	Interrupt clear Register
0x634-0x6FC	Reserved
0x700	Unicast Address word 0
0x704	Unicast Address word 1
0x708	Frame filter Control
0x70C	Frame filter Enable
0x710	Frame filter value bytes 3-0
0x714	Frame Filter value bytes 7-4
0x718	Frame Filter value bytes 11-8
0x71C	Frame Filter value bytes 15-12
0x720	Frame Filter value bytes 19-16
0x724	Frame Filter value bytes 23-20
0x728	Frame Filter value bytes 27-24
0x72C	Frame Filter value bytes 31-28
0x730	Frame Filter value bytes 35-32
0x734	Frame Filter value bytes 39-36
0x738	Frame Filter value bytes 43-40
0x73C	Frame Filter value bytes 47-44
0x740	Frame Filter value bytes 51-48
0x744	Frame Filter value bytes 55-52
0x748	Frame Filter value bytes 59-56
0x74C	Frame Filter value bytes 63-60
0x750	Frame filter mask value bytes 3-0
0x754	Frame Filter mask value bytes 7-4
0x758	Frame Filter mask value bytes 11-8
0x75C	Frame Filter mask value bytes 15-12
0x760	Frame Filter mask value bytes 19-16
0x764	Frame Filter mask value bytes 23-20
0x768	Frame Filter mask value bytes 27-24
0x76C	Frame Filter mask value bytes 31-28

Table 13: **Core Registers *(Cont'd)***

Address	Description
0x770	Frame Filter mask value bytes 35-32
0x774	Frame Filter mask value bytes 39-36
0x778	Frame Filter mask value bytes 43-40
0x77C	Frame Filter mask value bytes 47-44
0x780	Frame Filter mask value bytes 51-48
0x784	Frame Filter mask value bytes 55-52
0x788	Frame Filter mask value bytes 59-56
0x78C	Frame Filter mask value bytes 63-60
0x790-0x7FC	Reserved

Register Definition

Statistics Counters

The Statistics counters can be defined to be either 32 or 64 bits wide, with 64 bits being the default. When defined as 64 bits wide, the counter values are captured across two registers. In all cases a read of the lower 32-bit value causes the upper 32 bits to be sampled. A subsequent read of the upper 32-bit location returns this sampled value.

Note: If a different upper 32-bit location is read, an error is returned.

Table 14: **Statistics Counter Definitions**

Name	Type	Address	Description
Received bytes	RO	0x200-0x204	A count of bytes of frames received (destination address to frame check sequence inclusive).
Transmitted bytes	RO	0x208-0x20C	A count of bytes of frames transmitted (destination address to frame check sequence inclusive).
RX Undersize frames	RO	0x210-0x214	A count of the number of frames received that were fewer than 64 bytes in length but otherwise well formed.
RX Fragment frames	RO	0x218-0x21C	A count of the number of frames received that were fewer than 64 bytes in length and had a bad frame check sequence field.
RX 64 byte Frames	RO	0x220-0x224	A count of error-free frames received 64 bytes in length.
RX 65-127 byte Frames	RO	0x228-0x22C	A count of error-free frames received between 65 and 127 bytes in length.
RX 128-255 byte Frames	RO	0x230-0x234	A count of error-free frames received between 128 and 255 bytes in length.
RX 256-511 byte Frames	RO	0x238-0x23C	A count of error-free frames received between 256 and 511 bytes in length.
RX 512-1023 byte Frames	RO	0x240-0x244	A count of error-free frames received between 512 and 1023 bytes in length.
RX 1024-MaxFrameSize byte Frames	RO	0x248-0x24C	A count of error-free frames received between 1024 bytes and the specified IEEE 802.3-2008 maximum legal length.
RX Oversize Frames	RO	0x250-0x254	A count of otherwise error-free frames received that exceeded the maximum legal frame length specified in [Ref 1].
TX 64 byte Frames	RO	0x258-0x25C	A count of error-free frames transmitted that were 64 bytes in length.
TX 65-127 byte Frames	RO	0x260-0x264	A count of error-free frames transmitted between 65 and 127 bytes in length.

Table 14: **Statistics Counter Definitions *(Cont'd)***

Name	Type	Address	Description
TX 128-255 byte Frames	RO	0x268-0x26C	A count of error-free frames transmitted between 128 and 255 bytes in length.
TX 256-511 byte Frames	RO	0x270-0x274	A count of error-free frames transmitted between 256 and 511 bytes in length.
TX 512-1023 byte Frames	RO	0x278-0x27C	A count of error-free frames transmitted that were between 512 and 1023 bytes in length.
TX 1024-MaxFrameSize byte Frames	RO	0x280-0x284	A count of error-free frames transmitted between 1024 and the specified IEEE 802.3-2008 maximum legal length.
TX Oversize Frames	RO	0x288-0x28C	A count of otherwise error-free frames transmitted that exceeded the maximum legal frame length specified in [Ref 1].
RX Good Frames	RO	0x290-0x294	A count of error-free frames received.
RX Frame Check Sequence Errors	RO	0x298-0x29C	A count of received frames that failed the CRC check and were at least 64 bytes in length.
RX Good Broadcast Frames	RO	0x2A0-0x2A4	A count of frames successfully received and directed to the broadcast group address.
RX Good Multicast Frames	RO	0x2A8-0x2AC	A count of frames successfully received and directed to a non-broadcast group address.
RX Good Control Frames	RO	0x2B0-0x2B4	A count of error-free frames received that contained the special control frame identifier in the length/type field.
RX Length/ Type Out of Range	RO	0x2B8-0x2BC	A count of frames received that were at least 64 bytes in length where the length/type field contained a length value that did not match the number of MAC client data bytes received. The counter also increments for frames in which the length/type field indicated that the frame contained padding but where the number of MAC client data bytes received was greater than 64 bytes (minimum frame size). The exception is when the Length/Type Error Checks are disabled in the chosen MAC, in which case this counter will not increment.
RX Good VLAN Tagged Frames	RO	0x2C0-0x2C4	A count of error-free VLAN frames received. This counter will only increment when the receiver is configured for VLAN operation.
RX Good Pause Frames	RO	0x2C8-0x2CC	A count of error-free frames received that contained the MAC Control type identifier 88-08 in the length/type field, contained a destination address that matched either the MAC Control multicast address or the configured source address of the MAC, contained the PAUSE opcode and were acted upon by the MAC.
RX Bad Opcode	RO	0x2D0-0x2D4	A count of error-free frames received that contained the MAC Control type identifier 88-08 in the Length/Type field but were received with an opcode other than the PAUSE opcode.
TX Good Frames	RO	0x2D8-0x2DC	A count of error-free frames transmitted.
TX Good Broadcast Frames	RO	0x2E0-0x2E4	A count of error-free frames that were transmitted to the broadcast address.
TX Good Multicast Frames	RO	0x2E8-0x2EC	A count of error-free frames that were transmitted to a group destination address other than broadcast.
TX Good Underrun Errors	RO	0x2F0-0x2F4	A count of frames that would otherwise be transmitted by the core but could not be completed due to the assertion of TX_UNDERRUN during the frame transmission.
TX Good Control Frames	RO	0x2F8-0x2FC	A count of error-free frames transmitted that contained the MAC Control Frame type identifier 88-08 in the length/type field.

Table 14: **Statistics Counter Definitions *(Cont'd)***

Name	Type	Address	Description
TX Good VLAN Tagged Frames	RO	0x300-0x304	A count of error-free VLAN frames transmitted. This counter will only increment when the transmitter is configured for VLAN operation.
TX Good Pause Frames	RO	0x308-0x30C	A count of error-free PAUSE frames generated and transmitted by the MAC in response to an assertion of pause_req.
TX Single Collision Frames	RO	0x310-0x314	A count of frames involved in a single collision but subsequently transmitted successfully (half-duplex mode only).
TX Multiple Collision Frames	RO	0x318-0x31C	A count of frames involved in more than one collision but subsequently transmitted successfully (half-duplex mode only).
TX Deferred	RO	0x320-0x324	A count of frames whose transmission was delayed on its first attempt because the medium was busy (half-duplex mode only).
TX Late Collisions	RO	0x328-0x32C	A count of the times that a collision has been detected later than one slot Time from the start of the packet transmission. A late collision is counted twice - both as a collision and as a late Collision (half-duplex mode only).
TX Excess collisions	RO	0x330-0x334	A count of the frames that, due to excessive collisions, are not transmitted successfully (half-duplex mode only).
TX Excess Deferral	RO	0x338-0x33C	A count of frames that deferred transmission for an excessive period of time (half-duplex mode only).
TX Alignment Errors	RO	0x340-0x344	Asserted for received frames of size 64-bytes and greater which contained an odd number of received nibbles and which also contained an invalid FCS field.

Receiver Configuration

The register contents for the two receiver configuration words can be seen in Table 15 and Table 16.

Table 15: **Receiver Configuration Word 0 (0x400)**

Bit	Default Value	Type	Description
31-0	All 0s	RW	**Pause frame MAC Source Address[31:0]:** This address is used by the MAC to match against the destination address of any incoming flow control frames. It is also used by the flow control block as the source address (SA) for any outbound flow control frames. The address is ordered so the first byte transmitted/received is the lowest positioned byte in the register; for example, a MAC address of AA-BB-CC-DD-EE-FF would be stored in Address[47:0] as 0xFFEEDDCCBBAA. Tie to the same value as EMAC_UNICASTADDR[31:0].

Table 16: **Receiver Configuration Word 1 (0x404)**

Bit	Default Value	Type	Description
15-0	All 0s	RW	**Pause frame MAC Source Address[47:32]:** See description in Table 15. Tie to the same value as EMAC_UNICASTADDR[47:32].
23-16	N/A	RO	Reserved
24	0	RW	**Control Frame Length Check Disable:** When this bit is set to '1,' the core does not mark control frames as 'bad' if they are greater than the minimum frame length.
25	0	RW	**Length/Type Error Check Disable:** When this bit is set to '1,' the core does not perform the length/type field error checks as described in [Ref 3]. When this bit is set to '0,' the length/type field checks is performed: this is normal operation.
26	0	RW	**Half Duplex:** (applicable in 10/100 Mb/s mode only, for certain physical interfaces). When this bit is 1, the receiver operates in half-duplex mode. When the bit is 0, the receiver operates in full-duplex mode.

Table 16: **Receiver Configuration Word 1 (0x404) *(Cont'd)***

Bit	Default Value	Type	Description
27	0	RW	**VLAN Enable:** When this bit is set to '1,' VLAN tagged frames are accepted by the receiver.
28	1	RW	**Receiver Enable:** If set to '1,' the receiver block is operational. If set to '0,' the block ignores activity on the physical interface RX port.
29	0	RW	**In-band FCS Enable:** When this bit is '1,' the MAC receiver passes the FCS field up to the client as described in [Ref 3]. When it is '0,' the client is not passed to the FCS. In both cases, the FCS is verified on the frame.
30	0	RW	**Jumbo Frame Enable:** When this bit is set to '1,' the MAC receiver accepts frames over the specified IEEE 802.3-2008 maximum legal length. When this bit is '0,' the MAC only accepts frames up to the specified maximum.
31	0	RW	**Reset:** When this bit is set to '1,' the receiver is reset. The bit then automatically reverts to '0.' This reset also sets all of the receiver configuration registers to their default values.

Transmitter Configuration

The register contents for the Transmitter Configuration Word are described in Table 17.

Table 17: **Transmitter Configuration Word (0x408)**

Bit	Default Value	Type	Description
24-0	N/A	RO	Reserved
25	0	RW	**Interframe Gap Adjust Enable:** If '1,' the transmitter reads the value on the port tx_ifg_delay at the start of frame transmission and adjusts the interframe gap following the frame accordingly (see [Ref 3]). If '0,' the transmitter outputs a minimum interframe gap of at least twelve clock cycles, as specified in [Ref 1].
26	0	RW	**Half Duplex:** (applicable in 10/100 Mb/s mode only, for certain physical interfaces). If '1,' the transmitter operates in half-duplex mode.
27	0	RW	**VLAN Enable:** When this bit is set to '1,' the transmitter recognizes the transmission of VLAN tagged frames.
28	1	RW	**Transmit Enable:** When this bit is '1,' the transmitter is operational. When it is '0,' the transmitter is disabled.
29	0	RW	**In-band FCS Enable:** When this bit is '1,' the MAC transmitter expects the FCS field to be passed in by the client as described in [Ref 3]. When this bit is '0,' the MAC transmitter appends padding as required, computes the FCS and appends it to the frame.
30	0	RW	**Jumbo Frame Enable:** When this bit is set to '1,' the MAC transmitter sends frames that are greater than the specified IEEE 802.3-2008 maximum legal length. When this bit is '0,' the MAC only sends frames up to the specified maximum.
31	0	RW	**Reset:** When this bit is set to '1,' the transmitter is reset. The bit then automatically reverts to '0.' This reset also sets all of the transmitter configuration registers to their default values.

Flow Control Configuration

The register contents for the Flow Control Configuration Word are described in Table 18.

Table 18: **Flow Control Configuration Word (0x40C)**

Bit	Default Value	Type	Description
28-0	N/A	RO	Reserved
29	1	RW	**Flow Control Enable (RX):** When this bit is '1,' received flow control frames inhibits the transmitter operation as described in [Ref 3]. When this bit is '0,' received flow control frames are always passed up to the client.
30	1	RW	**Flow Control Enable (TX):** When this bit is '1,' asserting the pause_req signal sends a flow control frame out from the transmitter as described in [Ref 3]. When this bit is '0,' asserting the pause_req signal has no effect.
31	N/A	RO	Reserved

Ethernet MAC Mode Configuration

The register contents for the Ethernet MAC Mode Configuration Word are described in Table 19.

Table 19: **MAC Speed Configuration Word (0x410)**

Bits	Default Value	Type	Description
8-0	EMAC_LINKTIMERVAL[8:0]	RW	Sets the programmable link timer value, for operation with 1000BASE-X or SGMII modes
23-9	0	RO	Reserved
24	EMAC_RX16BITCLIENT_ENABLE	RO	**RX 16 Bit AXI4-Stream Enable**: When this bit is 1, the AXI4-Stream receive data interface is 16 bits wide. When this bit is 0, the receive data interface is 8 bits wide. This bit is valid only when using 1000BASE-X PCS/PMA mode.
25	EMAC_TX16BITCLIENT_ENABLE	RO	**TX 16 Bit AXI4-Stream Enable**: When this bit is 1, the AXI4-Stream transmit data interface is 16 bits wide. When this bit is 0, the transmit data interface is 8 bits wide. This bit is valid only when using 1000BASE-X PCS/PMA mode.
26	EMAC_HOST_ENABLE	RO	**MAC Management Enable**: When this bit is 1, the management interface is enabled. When this bit is 0, the management interface is disabled.
27	EMAC_1000BASEX_ENABLE	RW	**1000BASE-X Enable**: When this bit is 1, the Ethernet MAC is configured in 1000BASE-X mode
28	EMAC_SGMII_ENABLE	RW	**SGMII Enable**: When this bit is 1, the Ethernet MAC is configured in SGMII mode
29	EMAC_RGMII_ENABLE	RO	**RGMII Enable**: When this bit is 1, the Ethernet MAC is configured in RGMII mode.
31-30	{EMAC_SPEED_MSB, EMAC_SPEED_LSB}	RW	**MAC Speed Configuration** "00" - 10 Mb/s "01" - 100 Mb/s "10" - 1 Gb/s "11" - N/A

Note: The setting of the MAC Speed Configuration register is not affected by a reset.

RGMII/SGMII Configuration

The register contents for the RGMII/SGMII Configuration Word are described in Table 20.

Table 20: **RGMII/SGMII Configuration Word (0x420)**

Bits	Default Value	Type	Description
0	0	RO	**RGMII Link**: Valid in RGMII mode configuration only. When this bit is 1, the link is up. When this bit is 0, the link is down. This displays the link information from the PHY to the Ethernet MAC, encoded by GMII_RX_DV and GMII_RX_ER during the IFG.
1	0	RO	**RGMII Duplex Status**: Valid in RGMII mode configuration only. This bit is 0 for half-duplex and 1 for full-duplex. This displays the duplex information from the PHY to the Ethernet MAC, encoded by GMII_RX_DV and GMII_RX_ER during the IFG.
3-2	0	RO	**RGMII Speed**: Valid in RGMII mode configuration only. Link information from the PHY to the Ethernet MAC as encoded by GMII_RX_DV and GMII_RX_ER during the IFG. This two-bit vector is defined with the following values: • 10 = 1000 Mb/s • 01 = 100 Mb/s • 00 = 10 Mb/s • 11 = N/A
29-4	N/A	RO	Reserved
31-30	0	RO	**SGMII_Speed**: Valid in SGMII mode configuration only. This displays the SGMII speed information, as received by TX_CONFIG_REG[11:10] in the PCS/PMA register. This two-bit vector is defined with the following values: • 10 = 1000 Mb/s • 01 = 100 Mb/s • 00 = 10 Mb/s • 11 = N/A

ID Register

The register contents for the ID Register are described in Table 21.

Table 21: **ID Register (0x4F8)**

Bits	Default Value	Type	Description
7-0	0	RO	Patch Level (0- nopatch, 1-rev1 etc)
15-8	N/A	RO	Reserved
23-16	1	RO	Minor Rev
31-24	2	RO	Major Rev

Ability Register

The register contents for the Ability Register are described in Table 22

Table 22: **Ability Register (0x4FC)**

Bits	Default Value	Type	Description
0	1	RO	**10M Ability**: If set, the core is 10M capable
1	1	RO	**100M Ability**: If set, the core is 100M capable
2	1	RO	**1G Ability**: If set, the core is 1G capable
3-7	N/A	RO	Reserved
8	1	RO	Statistics Counters available
9	1	RO	Half duplex capable
10	1	RO	Frame Filter available
11-31	N/A	RO	Reserved

MDIO

Table 23 through Table 26 describe the registers used to access the MDIO interface.

MDIO Setup

The register contents for the MDIO Setup Word are described in Table 23.

Table 23: **MDIO Setup Word (0x500)**

Bits	Default Value	Type	Description
5-0	All 0s	RW	**Clock Divide[5:0]:** See [Ref 3].
6	0	RW	**MDIO Enable:** When this bit is '1,' the MDIO interface can be used to access attached PHY devices. When this bit is '0,' the MDIO interface is disabled and the MDIO signals remain inactive. A write to this bit only takes effect if Clock Divide is set to a non-zero value.
31-7	N/A	RO	Reserved

MDIO Control

The register contents for the MDIO Control Word are described in Table 24. See [Ref 3] for more detail.

Table 24: **MDIO Control Word (0x504)**

Bits	Default Value	Type	Description
6-0	N/A	RO	Reserved
7	0	RO	**MDIO ready:** When set the MDIO is enabled and ready for a new transfer. This is also used to identify when a previous transaction has completed (that is, Read data is valid)
10-8	N/A	RO	Reserved
11	0	WO	**Initiate:** Writing a 1 to this bit starts an MDIO transfer.
13-12	N/A	RO	Reserved
15-14	0	RW	**TX_OP:** This field controls the type of access performed when a one is written to initiate.
20-16	0	RW	**TX_REGAD:** This controls the register address being accessed.
23-21	N/A	RO	Reserved
28-24	0	RW	**TX_PHYAD:** This controls the PHY address being accessed.
31-29	N/A	RO	Reserved

The register contents for the MDIO Write Data are described in Table 25.

Table 25: **MDIO Write Data (0x508)**

Bits	Default Value	Type	Description
15-0	All 0s	RW	**Write Data**
31-16	N/A	RO	Reserved

MDIO Read Data

The register contents for the MDIO Read Data are described in Table 26.

Table 26: **MDIO Read Data (0x50C)**

Bits	Default Value	Type	Description
15-0	All 0s	RO	**Read Data:** Only valid when MDIO ready is sampled high.
16	0	RO	**MDIO Ready:** This is a copy of bit 7 of the MDIO Control Word.
31-17	N/A	RO	Reserved

Interrupt Control

Table 27 through Table 30 describes the registers used to access the Interrupt Controller. The only current interrupt source is MDIO ready. See [Ref 3] for more detail.

Interrupt Status Register

The register contents for the Interrupt Status Register are described in Table 27.

Table 27: **Interrupt status Register (0x600)**

Bits	Default Value	Type	Description
0	0	RO	Interrupt 0 Status
31-1	N/A	RO	Reserved

Interrupt Pending Register

The register contents for the Interrupt Pending Register are described in Table 28.

Table 28: **Interrupt Pending Register (0x610)**

Bits	Default Value	Type	Description
0	0	RO	Interrupt 0 Pending
10-8	N/A	RO	Reserved

Interrupt Enable Register

The register contents for the Interrupt Enable Register are described in Table 29.

Table 29: **Interrupt Enable Register (0x620)**

Bits	Default Value	Type	Description
0	0	RW	Interrupt 0 Enable
31-1	N/A	RO	Reserved

Interrupt Clear Register

The register contents for the Interrupt Clear Register are described in Table 30.

Table 30: **Interrupt Clear Register (0x630)**

Bits	Default Value	Type	Description
0	0	WO	Interrupt 0 Clear
10-8	N/A	RO	Reserved

Frame Filter Configuration

Table 31 through Table 36 describe the registers used to access the optional Frame Filter configuration when the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper is implemented with a frame filter. In addition to the unicast address, broadcast address and pause addresses, the frame filter can optionally be generated to respond to up to eight additional separate addresses. These are stored in an address table within the frame filter. See [Ref 3]. Table 33 through Table 36 show how the contents of the table are set.

If no frame filter is present, these registers do not exist and return 0s for a read from the stated addresses.

Unicast Address Configuration

The register contents for the two unicast address registers are described in Table 31 and Table 32.

Table 31: **Unicast Address (Word 0) (0x700)**

Bits	Default Value	Type	Description
31-0	unicast_address[31-0]	RW	**Frame filter unicast address[31:0]**: This address is used by the MAC to match against the destination address of any incoming frames. The address is ordered so the first byte transmitted/received is the lowest positioned byte in the register; for example, a MAC address of AA-BB-CC-DD-EE-FF would be stored in Address[47:0] as 0xFFEEDDCCBBAA.

Table 32: **Unicast Address (Word 1) (0x704)**

Bits	Default Value	Type	Description
15-0	unicast_address[47 downto 32]	RW	**Frame filter unicast address[47:32]**: See description in Table 31.
31-16	N/A	RO	Reserved

Frame Filter Control Register

The contents of the Frame Filter Control register are described in Table 33.

Table 33: **Frame Filter Control (0x708)**

Bits	Default Value	Type	Description
31	1	RW	**Promiscuous Mode:** If this bit is set to '1,' the Frame filter is set to operate in promiscuous mode. All frames are passed to the receiver client regardless of the destination address.
30-3	N/A	RO	Reserved
2-0	0	RW	**Filter Index:** All Frame filters are mapped to the same location with the filter index specifying which physical filter is to be accessed.

Frame Filter Enable Register

The contents of the Frame Filter Enable register are described in Table 34.

Table 34: **Frame Filter Enable (0x70C)**

Bits	Default Value	Type	Description
31-3	N/A	RO	Reserved
0	0	RW	**Filter Enable:** This enable relates to the physical Frame Filter pointed to by the Filter index. If clear, the filter passes all packets.

Frame Filter Value

The contents of the Frame Filter Value are described in Table 35 and Table 36.

Table 35: **Frame Filter Value (0x710-0x74C)**

Bits	Default Value	Type	Description
31-0	bits 47:0 =1 All other =0	RW	**Filter Value** All filter value registers have the same format. The lower 31 bits of filter value, at address 0x710, relating to the Filter at physical Frame **Filter index**, that is to be written to the address table. The value is ordered so that the first byte transmitted/received is the lowest positioned byte in the register; for example, a MAC address of AA-BB-CC-DD-EE-FF would be stored in Filter Value[47:0] as 0xFFEEDDCCBBAA. By default the frame filters are configured to match against the broadcast address.

Table 36: **Frame Filter Mask Value (0x750-0x78C)**

Bits	Default Value	Type	Description
31-0	bits 47:0 =1 All other =0	RW	**Mask Value** All mask value registers have the same format. If a mask bit is set to 1 then the corresponding bit of the Filter Value is compared by the frame filter. For example, if a basic Destination address comparison was desired then bits 47:0 should be written to 1 and all other bits to 0.

Verification

The Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper has been verified with extensive simulation, as detailed in this section.

Simulation

A highly parameterizable transaction-based test bench was used to test the core. Tests include:

- Register Access
- MDIO Access
- Frame Transmission and Error Handling
- Frame Reception and Error Handling
- Frame Filtering

Hardware Verification

The latest Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper has been designed to directly target the ML605 board, enabling the example design to be directly downloaded to demonstrate basic packet generation and loopback functionality if the XC6VLX240T-FF1156-1 part is targeted in the CORE Generator tool. This example design has been verified with the default configuration for all supported interface standards.

Device Utilization

Table 37 through Table 39 provide approximate utilization figures for various core options when a single instance of the core is instantiated.

Note: Virtex-6 devices support GMII and MII at 2.5V only; see the *Virtex-6 FPGA Data Sheet: DC and Switching Characteristics*.

Utilization figures are obtained by implementing the block level wrapper for the core.

Table 37 does not differentiate between GMII, MII and RGMII Physical Interfaces. The numbers quoted are for GMII 10/100/1000 Mb/s support.

BUFG usage:

- does not consider multiple instantiations of the core, where clock resources can often be shared.
- does not include the reference clock required for IDELAYCTRL. This clock source can be shared across the entire device and is not core specific.

Table 37: **Basic Device Utilization**

Core Parameters		Device Resources				
Physical Interface	**Management Interface**	**Slices**	**LUTs**	**FFs**	**LUTRAM**	**BUFGs**
MII/GMII/RGMII	AXI4	250	500	600	5	4
MII/GMII/RGMII	None	80	130	160	0	3
SGMII	AXI4	300	500	650	6	3
SGMII	None	80	120	200	1	2
SGMII [Elastic Buffer]	AXI4	400	650	900	60	3
SGMII [Elastic Buffer]	None	150	230	400	55	2
1000BASE-X (8-bit user AXI4-Stream Interface)	AXI4	300	500	650	6	3
1000BASE-X (8 bit user AXI4- Stream Interface)	None	80	120	200	1	2
1000BASE-X (16-bit user AXI4-Stream Interface)	AXI4	380	510	760	8	3
1000BASE-X (16-bit user AXI4-Stream Interface)	None	150	150	280	1	2

Additional Features

As well as the core utilization shown in Table 37, there are other features which can also be selected. Because the utilization of these features are not significantly affected by the core options they have been split out into separate tables.

Table 38: **Statistics Utilization**

Core Parameters		Device Resources			
Statistics Width	**Statistics Reset**	**Slices**	**LUTs**	**FFs**	**LUTRAM**
32	Yes	220	400	600	90
32	No	220	300	550	90
64	Yes	250	550	700	150
64	No	250	450	650	150

Table 39: **Frame Filter Utilization**

Core Parameters	Device Resources			
Filters	**Slices**	**LUTs**	**FFs**	**LUTRAM**
1	80	140	70	80
each additional filter	30	50	20	30

Performance

Table 40 specifies the maximum supported performance of the Ethernet MAC in various Virtex-6 devices. For a matrix of supported configurations, see [Ref 4].

Table 40: **Performance Capabilities**

Performance	Physical interface	Virtex-6 devices				Virtex-6 Lower Power devices
		LXT	SXT	HXT	CXT	
10 Mb/s	MII; or tri-speed GMII, RGMII, or SGMII	Yes	Yes	Yes	Yes	Yes
100 Mb/s		Yes	Yes	Yes	Yes	Yes
1000 Mb/s	GMII, RGMII	Yes	Yes	Limited(1)	Yes	Limited(2)
	SGMII, 1000BASE-X	Yes	Yes	Yes	Yes	Yes
Overclocked 2000 or 2500 Mb/s	1000BASE-X	Yes	Yes	Yes	Yes	No

Notes:

1. For some Virtex-6 HXT devices, use of the GMII or RGMII physical interface for 1 Gb/s operation will not meet the receiver setup and/or hold time requirements of the respective specification. Sufficient system margin and IODELAY tap settings are necessary for correct operation. See Xilinx Answer Record 40028 for more details. Performance in these devices improves, and might meet the specification, with higher speed grade parts.
2. See GMII and RGMII Physical Interface Performance in Virtex-6 Lower Power Devices.

GMII and RGMII Physical Interface Performance in Virtex-6 Lower Power Devices

Following are Ethernet MAC limitations for Virtex-6 Lower Power devices (-1L speed grades) when using the GMII or RGMII physical interface:

- Use of the GMII physical interface for 1 Gb/s operation will not meet the receiver setup and/or hold time requirements of the GMII specification by a total of at least 165 ps. Sufficient system margin and IODELAY tap settings are necessary for correct operation. See Xilinx Answer Record 40028 for more details.
- Use of the RGMII physical interface for 1 Gb/s operation is marginal with respect to the RGMII receiver timing specification. Sufficient system margin and IODELAY tap settings are necessary for correct operation. See Xilinx Answer Record 40028 for more details.

List of Acronyms

Acronym	Definition
AXI	Advanced eXtensible Interface
DA	Destination Address
DCM	Digital Clock Manager
DDR	Double Data Rate
FCS	Frame Check Sequence
FF	flip-flop
FIFO	First In First Out
FPGA	Field Programmable Gate Array
GBIC	Gigabit Interface Converter
Gb/s	Gigabits per second
GMII	Gigabit Media Independent Interface
HDL	Hardware Description Language
IO	Input/Output
IOB	Input/Output Block
IP	Intellectual Property
IPIF	IP Interface
ISE	Integrated Software Environment
LUT	Lookup Table
MAC	Media Access Controller
Mb/s	Megabits per second
MDIO	Management Data Input/Output
MII	Media Independent Interface
NGC	Native Generic Circuit
NGD	Native Generic Database
PCS	Physical Coding Sublayer
PHY	physical-side interface
PMA	Physical Medium Attachment
PMD	Physical Medium Dependent
RGMII	Reduced Gigabit Media Independent Interface
RO	Read Only
RW	Read Write
RX	Receive
SA	Source Address
SFP	Small Form-Factor Pluggable
SGMII	Serial Gigabit Media Independent Interface
TBI	Ten-Bit-Interface
TEMAC	Tri-Mode Ethernet MAC
TX	Transmit

Acronym	Definition
UCF	User Constraints File
VHDL	VHSIC Hardware Description Language (VHSIC an acronym for Very High-Speed Integrated Circuits)
VLAN	Virtual LAN (Local Area Network)
WO	Write Only

References

1. IEEE 802.3-2008 specification
2. Reduced Gigabit Media Independent Interface (RGMII), version 2.0
3. LogiCORE IP Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC User Guide (UG800)
4. Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC User Guide (UG368)
5. Xilinx AXI Design Reference Guide (UG761)
6. AMBA 4 AXI4-Stream Protocol Version: 1.0 Specification

Support

For technical support, visit www.xilinx.com/support. Xilinx provides technical support for this product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not listed in the documentation, if customized beyond that allowed in the product documentation, or if any changes are made in sections of design marked *DO NOT MODIFY*.

Ordering Information

The core is provided under the End User License Agreement and can be generated using CORE Generator software v13.3. The CORE Generator software is shipped with Xilinx ISE Design Suite software.

This core is provided at no charge and is shipped with the CORE Generator software with no separate license key required. See the Virtex-6 Embedded Tri-mode Ethernet MAC Wrapper product page for more details on this product. Contact your local Xilinx sales representative for pricing and availability about Xilinx LogiCORE IP modules and software or see the Xilinx IP Center. Information on additional LogiCORE IP modules is available on the Xilinx IP Center.

Revision History

Date	Version	Revision
3/1/11	1.1	Initial release for AXI support.
10/19/11	2.0	ISE Release 13.3. Addition of 1000BASE-X PCS/PMA overclocked modes and 16-bit AXI4-Stream Interface.

Notice of Disclaimer