LogiCORE IP Tri-Mode Ethernet MAC v5.2

DS818 January 18, 2012 **Product Specification**

Introduction

The LogiCORE™ IP Tri-Mode Ethernet Media Access Controller (TEMAC) solution comprises the 10/100/1000 Mb/s Ethernet MAC, 1 Gb/s Ethernet MAC and the 10/100 Mb/s Ethernet MAC IP core. All cores support half-duplex and full-duplex operation.

Features

- Designed to the *IEEE 802.3-2008* specification
- Configurable half-duplex and full-duplex operation
- Supports 10/100 Mb/s-only, 1 Gb/s-only or full 10/100/1000 Mb/s IP cores
- Supports RGMII, GMII and MII as well as providing connectivity to
  - LogiCORE IP Ethernet 1000BASE-X PCS/PMA or SGMII using transceiver, Select I/O or TBI
- Configured through an optional AXI4-Lite interface
- Configurable flow control through MAC Control pause frames
- Optional MDIO interface to managed objects in PHY layers (MII Management)
- Optional Frame Filter with selectable number of 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 statistics counters
- Optional Ethernet Audio Video Bridging (AVB) Endpoint designed to the following IEEE specifications
  - IEEE802.1AS
    Supports clock master functionality, clock slave functionality and the Best Master Clock Algorithm (BMCA)
  - IEEE802.1Qav
    Supports arbitration between different priority traffic and implements bandwidth policing
- Supports AXI4-Stream on RX and TX datapaths
- Available under the terms of the Xilinx Site License or Project License Agreement

LogiCORE IP Facts Table				
Core Specifics				
	Device			**Speed Grade**
Supported Device Family(1)	Virtex-7, Kintex-7, Artix-7, Virtex-6			-1
	Spartan-6(2)			-2
Supported User Interfaces	AXI4-Lite, AXI4-Stream			
Performance	10 Mb/s, 100 Mb/s, 1 Gb/s(3)			
Resources(4)				
Configuration	**LUTs**	**FFs**	**Slices**	**BUFG**
	600-3900	800-4200	350-2000	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			
Supported S/W Driver(5)	N/A			
Tested Design Tools				
Design Entry Tools	ISE® software v13.4			
Simulation(6)(7)	Mentor Graphics ModelSim Cadence Incisive Enterprise Simulator (IES) Synopsys VCS and VCS MX			
Synthesis Tools	XST 13.4			
Support				
Provided by Xilinx, Inc.				

1. For a complete listing of supported devices, see the release notes for this core.
2. Virtex-6 devices support GMII and MII at 2.5 V only; see [Ref 1] for more information. For Virtex-7, Kintex-7 and Artix-7 devices, it is I/O dependant with HR I/O supporting MII/GMII or RGMII at 2.5 V or lower and HP IO only supporting 1.8 V or lower.
3. Performance is subject to device support. See Performance.
4. See Table 55 to Table 57; precise number depends on user configuration and family.
5. Standalone driver details can be found in the EDK or SDK directory (*<install_directory>*/doc/usenglish/xilinx_drivers.htm). Linux OS and driver support information is available from http://wiki.xilinx.com.
6. Scripts provided for listed simulators only.
7. For the supported versions of the tools, see the ISE Design Suite 13: Release Notes Guide.

Applications

Typical applications for the TEMAC core include the following:

- Ethernet 1000BASE-X Port
- Ethernet Tri-Speed BASE-T Port (MII/GMII or RGMII)
- Ethernet Tri-Speed BASE-T Port (SGMII)
- Ethernet AVB Endpoint System

Ethernet 1000BASE-X Port

Figure 1 illustrates a typical 1 Gb/s MAC application. The TEMAC core can be generated with both 1 Gb/s only and full-duplex only to remove unnecessary logic. The PHY side of the core is connected to internally integrated serial transceivers, available in certain families, to connect to an external off-the-shelf Gigabit Interface Converter (GBIC) or Small Form-Factor Pluggable (SFP) optical transceiver. The 1000BASE-X logic can be provided by the Ethernet 1000BASE-X PCS/PMA or SGMII core.

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 TEMAC solution to complete a single Ethernet port. This port is shown connected to a Switch or Routing matrix, which can contain several ports.

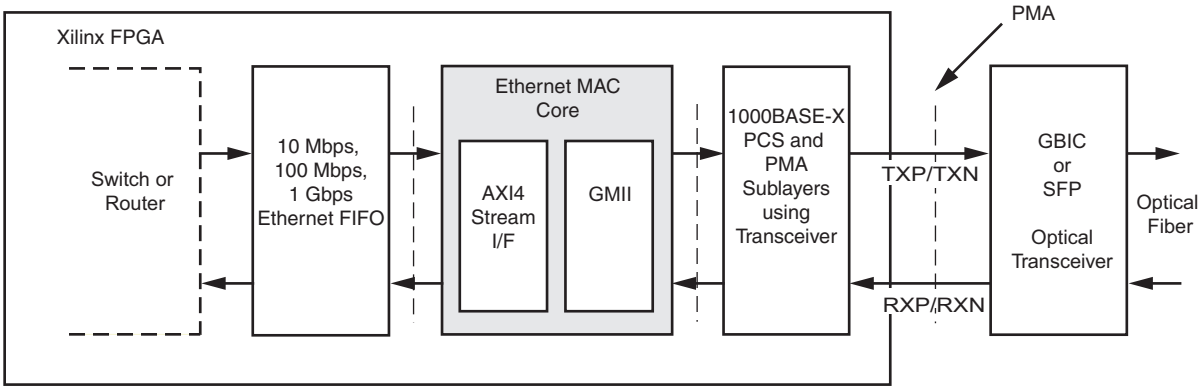

Figure 1: **Typical MAC 1000BASE-X Application**

Ethernet Tri-Speed BASE-T Port (MII/GMII or RGMII)

Figure 2 illustrates a typical application for the TEMAC (10/100/1000 Mb/s) core. The PHY side of the core is implementing an external GMII/MII. 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. Hardware Description Language (HDL) example designs are provided with the core to demonstrate external GMII or RGMII.

The user side of the TEMAC 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.

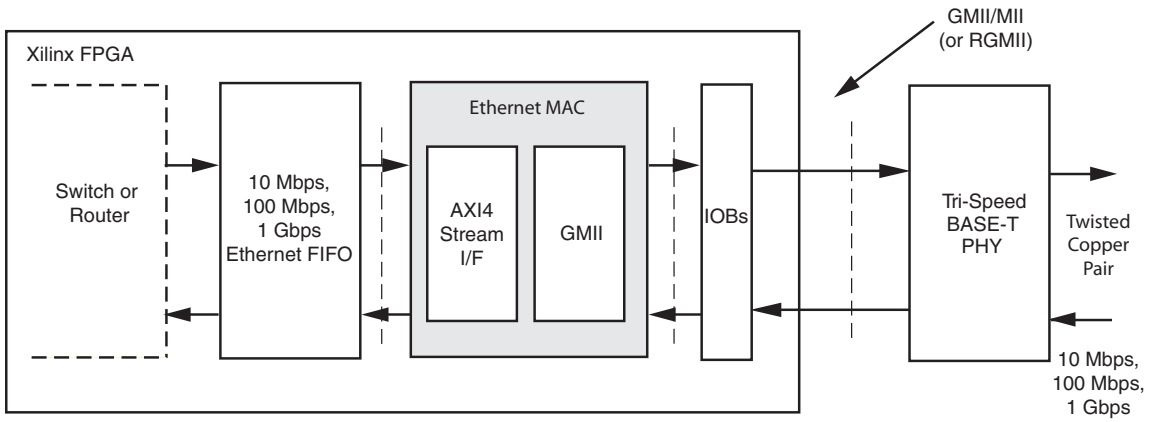

Figure 2: **Typical BASE-T Application for TEMAC Core: MII/GMII/RGMII**

Ethernet Tri-Speed BASE-T Port (SGMII)

Figure 3 illustrates a typical application for the TEMAC (10/100/1000 Mb/s) core. The PHY side of the core is connected to internally integrated SGMII logic using the device-specific transceiver to connect 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. The SGMII logic can be provided by the Ethernet 1000BASE-X PCS/PMA or SGMII core using transceivers. See [Ref 2] for more information.

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 TEMAC core, to complete a single Ethernet port. This port is shown connected to a Switch or Routing matrix, which can contain several ports.

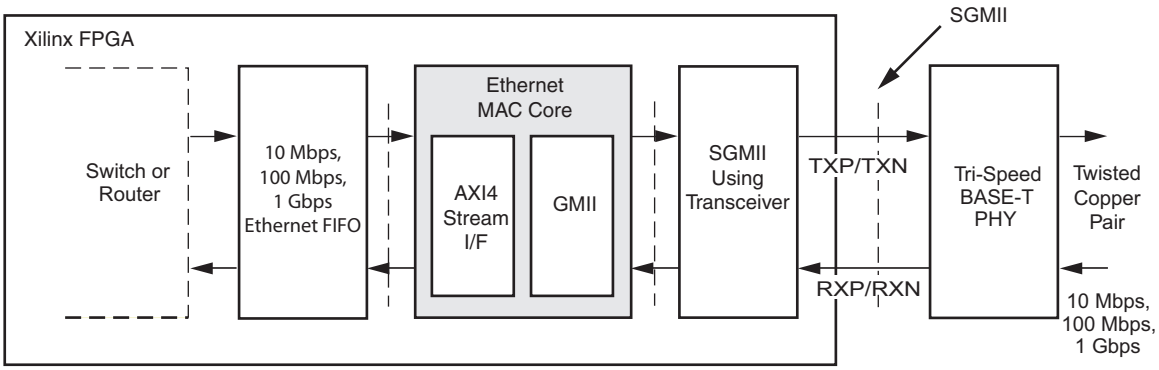

Figure 3: **Typical BASE-T Application for TEMAC Core: SGMII**

Ethernet AVB Endpoint System

Figure 4 illustrates a typical implementation for the TEMAC (100/1000 Mb/s) core when the optional Ethernet AVB endpoint is included. Endpoint refers to a talker or listener device as opposed to an intermediate bridge function, which is not supported. In the implementation, the TEMAC core, with the AVB front end, is connected to an AVB-capable network.

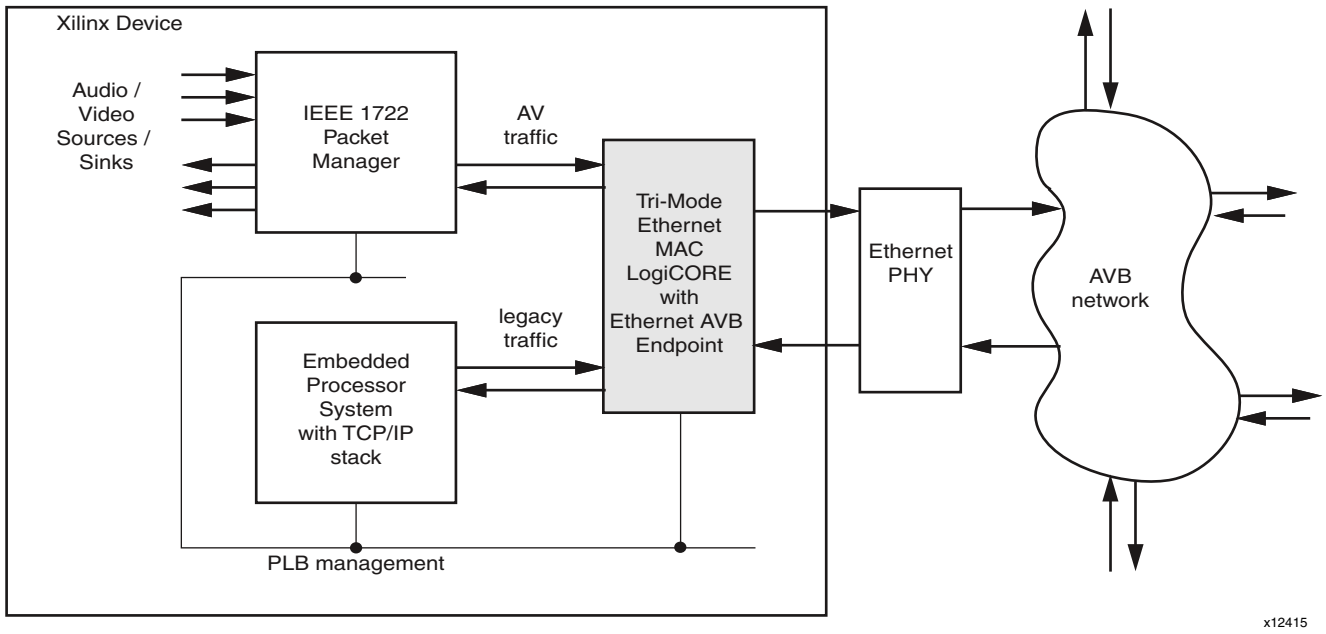

Figure 4: **Example Ethernet AVB Endpoint System**

Figure 4 illustrates that the TEMAC core with the Ethernet AVB Endpoint logic supports two main data interfaces at the user side:

1. The **AV traffic** interface is intended for the Quality of Service audio/video data. Illustrated are a number of audio/video sources (for example, a DVD player), and a number of audio/video sinks (for example, a TV set). The Ethernet AVB Endpoint gives priority to the **AV traffic** interface over the **legacy traffic** interface, as dictated by *IEEE 802.1Q* 75% bandwidth restrictions.
2. The **legacy traffic** interface is maintained for *best effort* Ethernet data: Ethernet as we know it today (for example, a PC surfing the internet). Wherever possible, priority is given to the **AV traffic** interface (as dictated by *IEEE 802.1Q* bandwidth restrictions), but a minimum of 25% of the total Ethernet bandwidth is always available for legacy Ethernet applications.

The **AV traffic** interface in Figure 4 is shown as interfacing to a 1722 Packet Manager block. The *IEEE1722* is another standard which specifies the embedding of audio/video data streams into Ethernet Packets. The 1722 headers within these packets include presentation timestamp information. Contact Xilinx for more system-level information.

Ethernet Architecture Overview

The MAC sublayer provided by this core is part of the Ethernet architecture displayed in Figure 5. The portion of the architecture, from the MAC to the right, is defined in [Ref 7]. This figure also illustrates where the supported interfaces fit into the architecture.

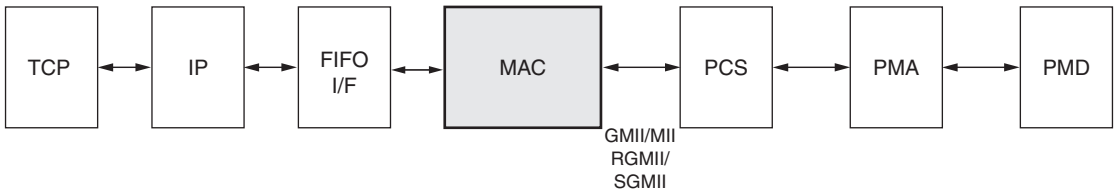

Figure 5: **Typical Ethernet Architecture**

MAC

The Ethernet Medium Access Controller (MAC) is defined in [Ref 7] clauses 2, 3, and 4. A MAC is responsible for the Ethernet framing protocols and error detection of these frames. The MAC is independent of, and can be connected to, any type of physical layer.

GMII / MII

The Gigabit Media Independent Interface (GMII) is defined in [Ref 7], clause 35. At 10 Mb/s and 100 Mb/s, the Media Independent Interface (MII) is used as defined in [Ref 7], clause 22. These are parallel interfaces connecting a MAC to the physical sublayers (PCS, PMA, and PMD).

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. No change in the operation of the core is required to select between GMII and RGMII. However, the clock management logic and Input/Output Block (IOB) logic around the core will change. HDL example designs are provided with the core which implement either the GMII or RGMII protocols.

SGMII

The Serial-GMII (SGMII) is an alternative interface to the GMII, which converts the parallel interface of the GMII into a serial format, radically reducing the I/O count (and for this reason often favored by PCB designers).

The TEMAC solution can be extended to include SGMII functionality by internally connecting its PHY side GMII to the Ethernet 1000BASE-X PCS/PMA or SGMII core from Xilinx. See [Ref 9].

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 fibre 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 2, these can be connected using GMII/MII, RGMII, or SGMII to provide a tri-speed Ethernet port.

The 1000BASE-X architecture illustrated in Figure 1 can be provided by connecting the TEMAC core to the Ethernet 1000BASE-X PCS/PMA or SGMII core.

Block Overview

Figure 6 identifies the major functional blocks of the TEMAC solution. Descriptions of the functional blocks and interfaces are provided in the subsequent sections.

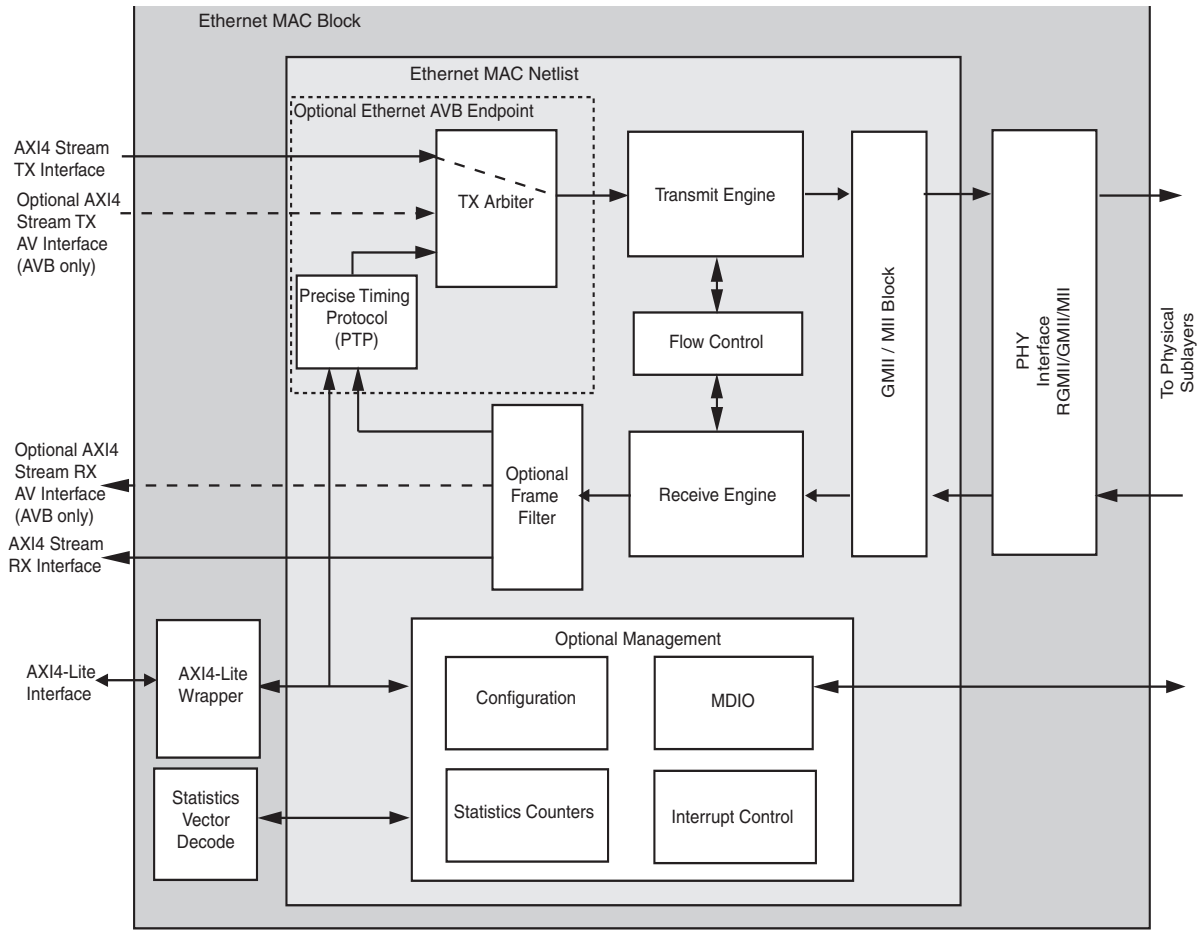

Figure 6: **TEMAC Functional Block Diagram**

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 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 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 using either RGMII or GMII/MII. The core can be generated without the PHY Interface to allow direct connection to the LogiCORE IP ethernet 1000BASE-X PCS/PMA or SGMII.

Ethernet AVB Endpoint

The TEMAC can be implemented with an optional Ethernet AVB endpoint which itself is made up of two key functional blocks. When this functionality is not included the AXI4-Stream TX Data is passed directly to the transmit engine. The AXI4-Stream RX Data is always passed directly to the user, with the relative `tuser` signals being used to validate the data on the required interface.

Precise Timing Protocol (PTP)

The Precise Timing Protocol (PTP) block within the core provides the dedicated hardware to implement the *IEEE 802.1AS* specification. However, full functionality is only achieved using a combination of this hardware block coupled with functions provided by the relevant software drivers (run on an embedded processor). For more information see [Ref 9].

TX Arbiter

Data for transmission over an AVB network can be obtained from three source types:

1. **AV Traffic.** For transmission from the AV Traffic I/F of the core.
2. **Precise Timing Protocol (PTP) Packets**. Initiated by the software drivers using the dedicated hardware
3. **Legacy Traffic**. For transmission from the Legacy Traffic I/F of the core.

The transmitter (Tx) arbiter selects from these three sources in the following manner. If there is an AV packet available and the programmed AV bandwidth limitation is not exceeded then the AV packet is transmitted; otherwise the Tx arbiter checks to see if there are any PTP packets to be transmitted and if there is an available legacy packet then this is transmitted. To comply with the specifications, the AV Traffic Interface should not be configured to exceed 75%. The arbiter then polices this bandwidth restriction for the AV traffic and ensures that on average, it is never exceeded. Consequently, despite the AV traffic having a higher priority than the legacy traffic, there is always remaining bandwidth available to schedule legacy traffic.

Transmit Engine

The transmit engine takes data from the AXI4-Stream TX interface and converts it to GMII format. Preamble and frame check sequence fields are added and the data is padded if necessary. The transmit engine also provides the transmit statistics vector for each packet and transmits the pause frames generated by the flow control module.

Receive Engine

The receive engine takes the data from the GMII/MII interface and checks it for compliance to [Ref 7]. Padding fields are removed and the AXI4-Stream RX interface is presented with the data along with a good/bad indication. The receive engine also provides the receive statistics vector for each received packet.

Flow Control

The flow control block is designed to [Ref 7], clause 31. The MAC can be configured to send pause frames with a programmable pause value and to act on their reception. These two behaviors can be configured asymmetrically.

GMII/MII Block

The GMII/MII interface, which only operates at speeds below 1 Gb/s, converts between the 4-bit data required by MII and the 8-bit data expected by the Receiver/Transmitter interfaces.

Management Interface

The optional Management Interface is a processor-independent interface with standard address, data, and control signals. It is used for the configuration and monitoring of the 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 can be configured using configuration vectors.

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 PHY devices. The MDIO Interface is defined in [Ref 7], clause 22.

Frame Filter

The TEMAC solution can be implemented with an optional Frame Filter. If the Frame Filter is enabled, the device does not pass frames that do not contain one of a set of known addresses or match against one of the configurable frame filters. By default, all configurable frame filters only match against the destination address when set to the broadcast address.

When the AVB Endpoint is included the Frame Filter is always present with three filters being dedicated to identifying AV or PTP data. In this case these filters are initialized to identify the default values for the various frame fields. The number of filters selected by the user is in addition to these three.

Statistics Counters

The TEMAC solution can be implemented with optional Statistics Counters. See [Ref 9] for more details.

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).

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 9].

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]	Input	tx_mac_aclk	Frame data to be transmitted.
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.

Note: All signals are active High.

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.
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.

Note: All signals are active High.

Table 3 defines the optional AXI4-Stream AV transmit signals included when the AVB functionality is selected.

Table 3: **Transmit Interface AXI4-Stream AV Signal Pins**

Signal	Direction	Clock Domain	Description
tx_axis_av_tdata[7:0]	Input	tx_mac_aclk	Frame data to be transmitted.
tx_axis_av_tvalid	Input	tx_mac_aclk	Control signal for tx_axis_av_tdata port. Indicates the data is valid.
tx_axis_av_tlast	Input	tx_mac_aclk	Control signal for tx_axis_av_tdata port. Indicates the final transfer in a frame.
tx_axis_av_tuser	Input	tx_mac_aclk	Control signal for tx_axis_av_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_av_tready	Output	tx_mac_aclk	Handshaking signal. Asserted when the current data on tx_axis_av_tdata has been accepted and tx_axis_av_tvalid is high. At 100Mb/s this is used to meter the data into the core at the correct rate.

Note: All signals are active High

Transmitter AXI4-Stream Interface Timing

Figure 7 displays a typical frame transmission at the user interface. All signals are synchronous to the `tx_mac_aclk` clock. See [Ref 9] 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.

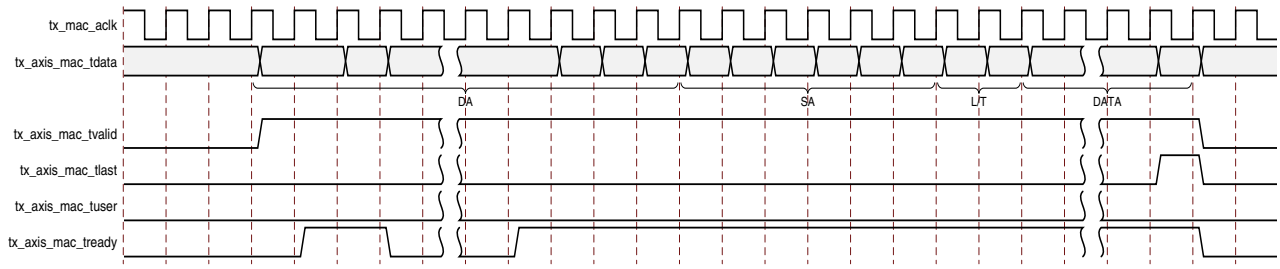

Figure 7: **Normal Frame Transmission across AXI4-Stream Interface**

Transmitter AXI4-Stream AV specifics

One of the key functions of the Ethernet AVB Endpoint is the configurable bandwidth allocation for the AV user data. Because this bandwidth is managed over time this is done using credits which are gained when non-AV data is sent and lost when AV data is sent, with a positive or zero balance of credits enabling the AV path. When no data is present at the AV input, any credits available are removed thus preventing bursty AV traffic getting an artificially high bandwidth. The `tx_axis_av_tvalid` indicates that data is available, but at the end of a frame, if another frame is available, the `tx_axis_av_tvalid` should remain asserted and the first byte of the new frame should be presented. This is shown in Figure 8. If `tx_axis_av_tvalid` is dropped between frames then any positive credit balance is lost whereas a negative balance remains, which results in a lower overall bandwidth allowance for the AV path.

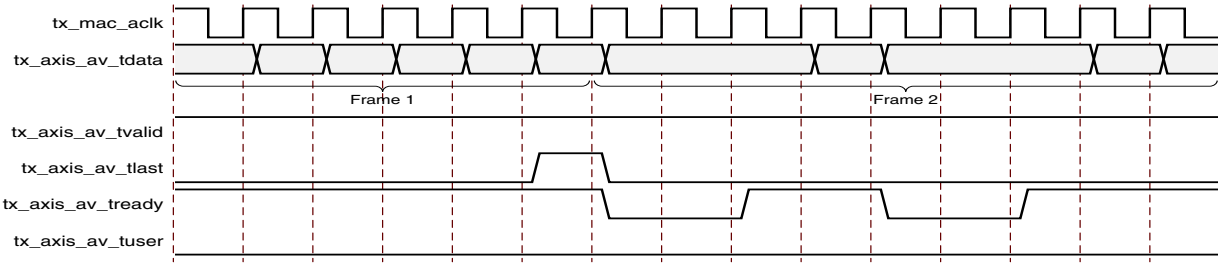

Figure 8: **TX AXI4 Stream AV Timing**

Receiver Interface

Signal Definition

Table 4 describes the receive AXI4-Stream signals used by the core to transfer data to the user. Table 5 describes the related sideband interface signals.

Table 4: **Receive Interface AXI4-Stream Signal Pins**

Signal	Direction	Clock Domain	Description
rx_axis_mac_tdata[7:0]	Output	rx_mac_aclk	Frame data received is supplied on this port.
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	Output	rx_mac_aclk	Control signal for the rx_axis_mac_tdata port. Indicates the final byte in the frame.
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 9] for more information.

Note: All signals are active High.

Table 5: **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.

Table 6 defines the optional AXI4-Stream AV receive signals included when the AVB functionality is selected.

Table 6: **Receive Interface AXI4-Stream AV Signal Pins**

Signal	Direction	Clock Domain	Description
rx_axis_av_tdata[7:0]	Output	rx_mac_aclk	Frame data received is supplied on this port.
rx_axis_av_tvalid	Output	rx_mac_aclk	Control signal for the rx_axis_av_tdata port. Indicates the data is valid.
rx_axis_av_tlast	Output	rx_mac_aclk	Control signal for the rx_axis_av_tdata port. Indicates the final byte in the frame.
rx_axis_av_tuser	Output	rx_mac_aclk	Control signal for rx_axis_av_tdata. Asserted at end of frame reception to indicate that the frame had an error.

Note: All signals are active High.

Receiver AXI4-Stream Interface Timing

Figure 9 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 or did not match the appropriate filter, for example, all frames are present on both the rx_axis_mac interface and the rx_axis_av interface with the relative tuser being used to drop the frame from the unintended interface.

Note: The core does not have any way to throttle the data; it is assumed that the data can be taken 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.

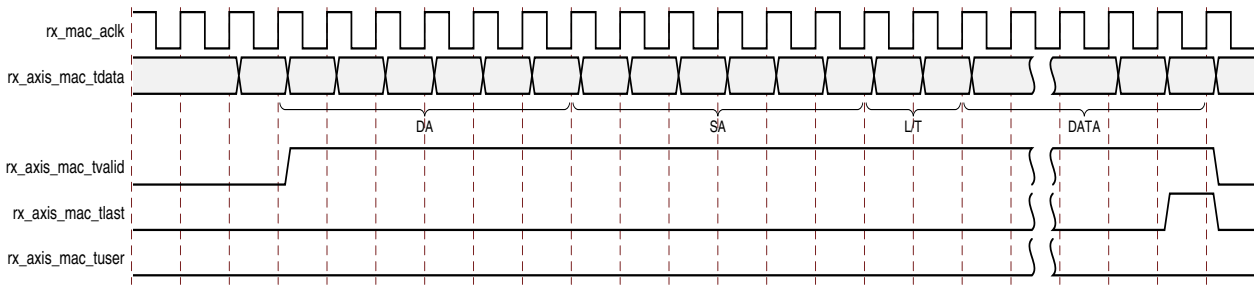

Figure 9: **Normal Frame Reception at AXI4-Stream Interface**

Receiver AXI4-Stream AV Interface Timing

The RX AV AXI4-Stream interface has the same timing as shown in Figure 9, with all signal names replacing axis_mac with axis_av.

Flow Control User Side Interface Signal Definition

Table 7 describes the signals used by the user 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 7]. 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 7: **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 8 describes the optional signals used by the user to access the MAC netlist, including configuration, status and MDIO access.

Table 8: **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

Configuration Vector Signal Definition

Table 9 describes the configuration vectors, which use direct inputs to the core to replace the functionality of the MAC configuration bits when the Management Interface is not used. The configuration settings described in Table 19 through Table 25 are included in the vector. See [Ref 9] for detailed information.

Table 9: **Alternative to the Optional Management Interface: Configuration Vector Signal Pinout**

Signal	Direction	Description
rx_mac_config_vector[79:0]	Input	The RX Configuration Vector is used to replace the functionality of the MAC RX Configuration Registers when the Management Interface is not used.
tx_mac_config_vector[79:0]	Input	The TX Configuration Vector is used to replace the functionality of the MAC TX Configuration Registers when the Management Interface is not used.

Note: All bits of the config vectors are registered on input but can be treated as asynchronous inputs.

Clock, Speed Indication, and Reset Signal Definition

Table 10 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 10: **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

Table 10: **Clock and Speed Indication Signals *(Cont'd)***

Signal	Direction	Description
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 125 MHz clock
rtc_clk	Input	Only available when the core is generated with AVB. Reference clock used to increment the Real Time Clock (RTC). The minimum frequency is 25 MHz. Xilinx recommends a 125 MHz clock source shared with gtx_clk.
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 (if the optional Management Interface is present) or from the configuration vector (if the optional Management Interface is not present).
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 (if the optional Management Interface is present) or from the configuration vector (if the optional Management Interface is not present).

Interrupt Signals

Table 11 describes the interrupt signals provided by the TEMAC core.

Table 11: **Interrupt signals**

Signal	Direction	Description
mac_int	Output	This is the interrupt output from the interrupt controller. Currently the only interrupt source which can be configured is the mdio_ready signal. See [Ref 9] for more information.
interrupt_ptp_rx	Output	Only available when the core is generated with AVB. This is asserted following the reception of any PTP packet by the RX PTP Packet Buffers. See [Ref 9] for more information.
interrupt_ptp_tx	Output	Only available when the core is generated with AVB. This is asserted following the transmission of any PTP packet from the TX PTP Packet Buffers. See [Ref 9] for more information.
interrupt_ptp_timer	Output	Only available when the core is generated with AVB. This interrupt asserts every 1/128 seconds as measured by the RTC. This acts as a timer for the PTP software algorithms. See [Ref 9] for more information.

Ethernet AVB Endpoint PTP Signals

Table 12 defines the signals output from the core by the Precise Timing Protocol (PTP). These signals, present only when the AVB Endpoint is included in the TEMAC, are provided for reference only and can be used by an application.

Table 12: **AVB Specific signals**

Signal	Direction	Description
rtc_nanosec_field	Output	This is the synchronised nanoseconds field from the RTC
rtc_sec_field	Output	This is the synchronised seconds fields from the RTC

Table 12: **AVB Specific signals *(Cont'd)***

Signal	Direction	Description
clk8k	Output	This is an 8 kHz clock which is derived from, and synchronized in frequency, to the RTC. The period of this clock, 125 µs, can be useful in timing SR class measurement intervals.
rtc_nanosec_field_1722	Output	The IEEE1722 specification contains a different format for the RTC, provided here as an extra port. This is derived and is synchronous with the IEEE802.1 AS RTC.

Physical Interface Signal Definition

Table 13 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 7], clause 22.

Table 13: **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.
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 14 through Table 16 describe the three possible interface standards supported, RGMII, GMII and MII, which are typically attached to a PHY module, either off-chip or internally integrated. The RGMII is defined in [Ref 8], the GMII is defined in[Ref 7], clause 35, and MII is defined in [Ref 7], clause 22.

Table 14: **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 15: **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

Table 15: **Optional MII Interface Signal Pinout *(Cont'd)***

Signal	Direction	Clock Domain	Description
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 16: **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
inband_link_status	Output	rgmii_rxc	Link Status from the PHY
inband_clock_speed	Output	rgmii_rxc	Link Speed from the PHY
inband_duplex_status	Output	rgmii_rxc	Duplex Status from the PHY

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 parameters can be controlled via the configuration vectors as described in Configuration Vector Signal Definition. 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 17.

Table 17: **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

Table 17: **Core Registers *(Cont'd)***

Address	Description
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
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)

Table 17: **Core Registers *(Cont'd)***

Address	Description
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
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

Table 17: **Core Registers *(Cont'd)***

Address	Description
0x40C	Flow Control Configuration
0x410	Speed configuration
0x414	RX Max Frame Configuration
0x418	TX Max Frame Configuration
0x41C-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
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

Table 17: **Core Registers *(Cont'd)***

Address	Description
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
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
0x800-0xFFFC	Reserved
0x10000-0x100FC	RX PTP Buffer 0
0x10100-0x101FC	RX PTP Buffer 1
0x10200-0x102FC	RX PTP Buffer 2
0x10300-0x103FC	RX PTP Buffer 3
0x10400-0x104FC	RX PTP Buffer 4
0x10500-0x105FC	RX PTP Buffer 5
0x10600-0x106FC	RX PTP Buffer 6
0x10700-0x107FC	RX PTP Buffer 7
0x10800-0x108FC	RX PTP Buffer 8
0x10900-0x109FC	RX PTP Buffer 9
0x10A00-0x10AFC	RX PTP Buffer 10
0x10B00-0x10BFC	RX PTP Buffer 11
0x10C00-0x10CFC	RX PTP Buffer 12
0x10D00-0x10DFC	RX PTP Buffer 13

Table 17: **Core Registers *(Cont'd)***

Address	Description
0x10E00-0x10EFC	RX PTP Buffer 14
0x10F00-0x10FFC	RX PTP Buffer 15
0x11000-0x110FC	TX PTP Buffer 0
0x11100-0x111FC	TX PTP Buffer 1
0x11200-0x112FC	TX PTP Buffer 2
0x11300-0x113FC	TX PTP Buffer 3
0x11400-0x114FC	TX PTP Buffer 4
0x11500-0x115FC	TX PTP Buffer 5
0x11600-0x116FC	TX PTP Buffer 6
0x11700-0x117FC	TX PTP Buffer 7
0x11800-0x11FFC	Reserved
0x12000	TX PTP Packet Buffer Control Register
0x12004	RX PTP Packet Control Register
0x12008	Reserved
0x1200C	TX Arbiter Send Slope control Register
0x12010	TX Arbiter Idle Slope control Register
0x12014-0x127FC	Reserved
0x12800	RTC Nano-seconds Field Offset
0x12804	Reserved
0x12808	RTC Seconds Field Offset [31:0]
0x1280C	RTC Seconds Field Offset [47:32]
0x12810	RTC Increment Value Control Register
0x12814	Current RTC Nanoseconds Value
0x12818	Current RTC Seconds Value Bits [31:0]
0x1281C	Current RTC Seconds Value Bits [47:32]
0x12820	RTC Interrupt Clear Register
0x12824	RTC Phase Adjustment Register
0x12828-0x13FFC	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 18: **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 *IEEE 802.3-2008*.
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.
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 *IEEE 802.3-2008*.
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.

Table 18: **Statistics Counter Definitions *(Cont'd)***

Name	Type	Address	Description
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.
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 19 and Table 20.

Table 19: **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.

Table 20: **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 19.
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 9]. When this bit is set to '0,' the length/type field checks is performed: this is normal operation.
26	0	RW	**Half Duplex:** If '1,' the receiver operates in half- duplex mode. If '0,' the receiver operates in full-duplex mode.
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 9]. 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 21.

Table 21: **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 9]). If '0,' the transmitter outputs a minimum interframe gap of at least twelve clock cycles, as specified in *IEEE 802.3-2008*.
26	0	RW	**Half Duplex:** 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.

Table 21: **Transmitter Configuration Word (0x408) *(Cont'd)***

Bit	Default Value	Type	Description
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 9]. 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 22.

Table 22: **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 9]. 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 9]. When this bit is '0,' asserting the pause_req signal has no effect.
31	N/A	RO	Reserved

TEMAC Speed Configuration

The register contents for the MAC Speed Configuration Word, when the TEMAC solution has been generated with tri-speed support, are described in Table 23.

When the TEMAC solution has been generated for only 1 Gb/s speed support, bits 31-30 are hard-coded to the value '10'.

When the TEMAC solution has been generated for only 10 Mb/s and 100 Mb/s speed support, bits 31-30 only accept the values of '00' to configure for 10 Mb/s operation, or '01' to configure for 100 Mb/s operation.

Table 23: **MAC Speed Configuration Word (0x410)**

Bits	Default Value	Type	Description
29-0	N/A	RO	Reserved
31-30	10	RW	**MAC Speed Configuration** 00 - 10 Mb/s 01 - 100 Mb/s 10 - 1 Gb/s

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

RX Max Frame Configuration

The register contents for the Rx Max Frame Configuration Word are described in Table 24.

Table 24: **RX Max Frame Configuration Word (0x414)**

Bits	Default Value	Type	Description
14-0	0x5EE	RW	**RX Max Frame Length**
15	N/A	RO	Reserved
16	0	RW	**RX Max Frame Enable:** When low, the MAC assumes use of the standard 1518/1522 depending upon the setting of **VLAN enable**. When high, the MAC allows frames up to **RX Max Frame Length** irrespective of the value of **VLAN enable**. If **Jumbo Enable** is set then this register has no effect.
31-17	N/A	RO	Reserved

TX Max Frame Configuration

The register contents for the Tx Max Frame Configuration Word are described in Table 25.

Table 25: **TX Max Frame Configuration Word (0x418)**

Bits	Default Value	Type	Description
14-0	0x5EE	RW	**TX Max Frame Length**
15	N/A	RO	Reserved
16	0	RW	**TX Max Frame Enable:** When low the MAC assumes use of the standard 1518/1522 depending upon the setting of **VLAN enable**. When high the MAC allows frames up to **TX Max Frame Length** irrespective of the value of **VLAN enable**. If **Jumbo Enable** is set then this register has no effect.
31-17	N/A	RO	Reserved

ID Register

The register contents for the ID Register are described in Table 26.

Table 26: **ID Register (0x4F8)**

Bits	Default Value	Type	Description
7-0	0	RO	Patch Level (0-No patch, 1-Rev1)
15-8	N/A	RO	Reserved
23-16	1	RO	Minor Rev
31-24	5	RO	Major Rev

Ability Register

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

Table 27: **Ability Register (ox4FC)**

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 28 through Table 31 describe the registers used to access the MDIO interface.

MDIO Setup

The register contents for the MDIO Setup Word are described in Table 28.

Table 28: **MDIO Setup Word (0x500)**

Bits	Default Value	Type	Description
5-0	All 0s	RW	**Clock Divide[5:0]:** See [Ref 9].
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 29. See [Ref 9] for more detail.

Table 29: **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

MDIO Write Data

The register contents for the MDIO Write Data are described in Table 30.

Table 30: **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 31.

Table 31: **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 32 through Table 35 describes the registers used to access the Interrupt Controller. The only current interrupt source is MDIO ready. See [Ref 9] for more detail.

Interrupt Status Register

The register contents for the Interrupt Status Register are described in Table 32.

Table 32: **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 33.

Table 33: **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 34.

Table 34: **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 35.

Table 35: **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 36 through Table 41 describe the registers used to access the optional Frame Filter configuration when the TEMAC solution 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 9]. Table 38 through Table 41 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 36 and Table 37.

Table 36: **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 37: **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 36.
31-16	N/A	RO	Reserved

Frame Filter Control Register

The contents of the Frame Filter Control register are described in Table 38.

Table 38: **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.
30-9	N/A	RO	Reserved
8	0	RW	**AVB Select**: If the AVB Endpoint is present this is used to indicate that the filter to be selected is one of the three dedicated filters.
7-3	N/A	RO	Reserved
2-0	0	RW	**Filter Index:** All Frame filters are mapped to the same location with the filter index and AVB Select specifying which physical filter is to be accessed. When an AVB filter is being selected only indexes of 0-2 are allowed.

Frame Filter Enable Register

The contents of the Frame Filter Enable register are described in Table 39.

Table 39: **Frame Filter Enable (0x70C)**

Bits	Default Value	Type	Description
31-3	N/A	RO	Reserved
0	1	RW	**Filter Enable:** This enable relates to the physical Frame Filter pointed to by the Filter index and take the value of AVB Select into account. If clear, the filter passes all packets.

Frame Filter Value

The contents of the Frame Filter Value are described in Table 40 and Table 41.

Table 40: **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 41: **Frame Filter Mask Value (0x750-0x790)**

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.

AVB Endpoint Configuration

Table 42 to Table 54 describe the registers used to access the optional AVB Endpoint functionality.

Rx PTP Packet Buffer Address Space

The Address space of the RX PTP Packet Buffers is 4k bytes in total. This represents the size of a single FPGA block RAM pair (4k bytes). Every byte of this block RAM can be read.

This address space is divided equally into 16 separate buffers of 256 bytes, each of which is capable of storing a unique PTP frame. When received, a PTP frame is written into one of these buffers; then the buffer write pointer increments and points to the next buffer in preparation for subsequent PTP frame reception.

Within each buffer, the entire PTP frame is written in (from MAC Destination Address through to the last byte from the data field), starting at the base address of that buffer. Following PTP frame reception, the RX timestamp captured for that frame is written into the top 4 bytes of the buffer used.

Tx PTP Packet Buffer Address Space

The Address space of the TX PTP Packet Buffers is 2k bytes in total, representing the size of a single FPGA block 18k RAM. Every byte of this block RAM is accessible by the CPU. This address space is divided equally into 8 separate buffers of 256 bytes, each of which is capable of storing a unique PTP frame: 7 of these buffer locations are pre-initialized with standard PTP frame syntax; however, each byte can be modified if desired.

Within each single buffer, the initial byte is used as a length field, used to indicate to the core logic the number of bytes to be transmitted for that frame. An entire PTP frame (from MAC Destination Address through to the last byte from the data field) is then stored, starting at the 8th address of that particular buffer. Following PTP frame transmission from one of these buffers, the TX Timestamp captured for that frame is written into the top 4 bytes of the buffer just used.

Tx PTP Packet Control Register

Table 42 defines associated control register of the TX PTP Packet Buffers, used by the software to request the transmission of the PTP frames.

Table 42: **Tx PTP Packet Buffer Control Register (0x12000)**

Bits	Default Value	Type	Description
7-0	0	WO	**tx_send_frame Bits**. The Tx PTP Packet Buffer is split into 8 regions of 256 bytes, each of which can contain a separate PTP frame. There is 1 tx_send_frame bit for each of the 8 regions. Each bit, when written to '1', causes a request to be made to the TX Arbiter. When access is granted the frame contained within the respected region is transmitted. If read, always returns 0.
15-8	0	RO	**tx_frame_waiting Indication**. The Tx PTP Packet Buffer is split into 8 regions of 256 bytes, each of which can contain a separate PTP frame. There is 1 tx_frame_waiting bit for each of the 8 regions. Each bit, when logic 1, indicates that a request has been made for frame transmission to the Tx Arbiter, but that a grant has not yet occurred. When the frame has been successfully transmitted, the bit is set to logic 0. This bit allows the microprocessor to run off a polling implementation as opposed to the Interrupts.
18-16	0	RO	**tx_packet**. Indicates the number (block RAM bin position) of the most recently transmitted PTP packet.
31-19	0	RO	Reserved

Note: A read or a write to this register clears the interrupt_ptp_tx interrupt (asserted after each successful PTP packet transmission).

Rx PTP Packet Control Register

Table 43 defines the associated control register of the RX PTP Packet Buffers used by the software to monitor the position of the most recently received PTP frame.

Table 43: **Rx PTP Packet Buffer Control Register (0x12004)**

Bits	Default Value	Type	Description
0	0	WO	**rx_clear**. When written with a '1,' forces the buffer to empty, in practice moving the write address to the same value as the read address. If read, always returns 0.
7-1	0	RO	Reserved
11-8	0	RO	**rx_packet**. Indicates the number (block RAM bin position) of the most recently received PTP packet.
31-12	0	RO	Reserved

Note: A read or a write to this register clears the `i`nterrupt_ptp_rx interrupt (asserted after each successful PTP packet reception).

Tx Arbiter Send Slope Control Register

The SendSlope variable is defined in IEEE802.1Qav-2009 to be the rate of change of credit, in bits per second, when the value of credit is decreasing (during AV packet transmission). Together with Tx Arbiter Idle Slope Control Register, RTC Nano-seconds Field Offset Control and RTC Seconds Field Offset Control, these registers define the maximum limit of the bandwidth reserved for AV traffic, as enforced by the TX Arbiter. The default values allow the maximum bandwidth proportion of 75% for the AV traffic. See the [Ref 7] or the [Ref 9] for further information.

Table 44: **Tx Arbiter Send Slope Control Register (0x1200C)**

Bits	Default Value	Type	Description
31-20	0	RO	Reserved
19-0	2048	R/W	The value of sendSlope

Tx Arbiter Idle Slope Control Register

The idleSlope variable is defined in *IEEE802.1Qav-2009* to be the rate of change of credit, in bits per second, when the value of credit is increasing (whenever there is no AV packet transmission). Together with Tx Arbiter Send Slope Control Register, RTC Nano-seconds Field Offset Control, and RTC Seconds Field Offset Control, these registers define the maximum limit of the bandwidth reserved for AV traffic: this is enforced by the TX Arbiter. The default values allow the maximum bandwidth proportion of 75% for the AV traffic. See [Ref 7] or [Ref 9] for further information.

Table 45: **Tx Arbiter Idle Slope Control Register (0x12010)**

Bits	Default Value	Type	Description
31-20	0	RO	Reserved
19-0	6144	R/W	The value of idleSlope

RTC Nano-seconds Field Offset Control

Table 46 describes the offset control register for the nano-seconds field of the RTC used to force step changes into the counter. When in PTP clock master mode, this can be used to set the initial value following power-up. When in PTP clock slave mode, the software drivers use this register to implement the periodic step corrections.

This register and the registers defined in Table 47 and in Table 48 are linked. These three offset values are loaded into the RTC counter logic simultaneously following a write to this nanosecond offset register.

Table 46: **RTC Nano-seconds Field Offset (0x12800)**

Bits	Default Value	Type	Description
29-0	0	R/W	**30-bit offset value for the RTC nano seconds**. Used by the microprocessor to initialize the RTC, then afterwards to perform the regular RTC corrections (when in slave mode).
31-30	0	RO	Reserved

RTC Seconds Field Offset Control

Table 47 describes the offset control register for the lower 32-bits of seconds field of the RTC, used to force step changes into the counter. When in PTP clock master mode, this can be used to set the initial value following power-up. When in PTP clock slave mode, the software drivers use this register to implement the periodic step corrections.

This register and the registers defined in Table 46 and in Table 48 are linked. These three offset values are loaded into the RTC counter logic simultaneously following a write to the nanosecond offset register defined in Table 46.

Table 47: **Seconds Field Offset Bits [31:0] (0x12808)**

Bits	Default Value	Type	Description
31-0	0	R/W	**32-bit offset value for the RTC seconds field (bits 31-0)**. Used by the microprocessor to initialize the RTC, then afterwards to perform the regular RTC corrections (when in slave mode).

Table 48 describes the offset control register for the upper 16-bits of seconds field of the RTC, used to force step changes into the counter. When in PTP clock master mode, this can be used to set the initial value following power-up. When in PTP clock slave mode, the software drivers use this register to implement the periodic step corrections.

This register and the registers defined in Table 46 and in Table 47 are linked. These three offset values are loaded into the RTC counter logic simultaneously following a write to the nanosecond offset register defined in Table 46.

Table 48: **Seconds Field Offset Bits [47:32] (0x1280C)**

Bits	Default Value	Type	Description
15-0	0	R/W	**16-bit offset value for the RTC seconds field (bits 47-32)**. Used by the microprocessor to initialize the RTC, then afterwards to perform the regular RTC corrections (when in slave mode).
31-16	0	RO	Reserved

RTC Increment Value Control Register

Table 49 describes the RTC Increment Value Control Register, which provides a configurable increment rate for the RTC counter. This increment register should take the value of the clock period being used to increment the RTC; however, the resolution of this increment register is very fine (in units of 1/1048576 ($1/2^{20}$) fraction of one nanosecond) and for this reason the RTC increment rate can be adjusted to a very fine degree of accuracy, thus providing the following features:

- The RTC can be incremented from any available clock frequency that is greater than the IEEE802.1AS defined minimum of 25 MHz.
- When acting as a clock slave, the rate adjustment of the RTC can be matched to that of the network clock master to an exceptional level of accuracy.

Table 49: **RTC Increment Value Control Register (0x12810)**

Bits	Default Value	Type	Description
25-0	0	R/W	Per rtc_clk clock period Increment Value for the RTC.
31-26	0	RO	Reserved

Current RTC Value Registers

Table 50 describes the nanoseconds field value register for the nano-seconds field of the RTV. When read, this returns the latest value of the counter. This register and the registers defined in Table 51 and in Table 52 are linked. When this nanoseconds value register is read, the entire RTC (including the seconds field) is sampled.

Table 50: **Current RTC Nanoseconds Value (0x12814)**

Bits	Default Value	Type	Description
29-0	0	RO	Current Value of the synchronized RTC nanoseconds field. Note: A read from this register samples the entire RTC counter (synchronized) so that the Epoch and Seconds field are held static for a subsequent read.
31-30	0	RO	Reserved

Table 51 describes the lower 32-bits of the seconds value register for the seconds field of the RTC. When read, this returns the latest value of the counter. This register and the registers defined in Table 50 and in Table 52 are linked. When the nanoseconds value register is read (see Table 50), the entire RTC is sampled.

Table 51: **Current RTC Seconds Field Value bits [31:0] (0x12818)**

Bits	Default Value	Type	Description
31-0	0	RO	Sampled Value of the synchronized RTC Seconds field (bits 31-0).

Table 52 describes the upper 16-bits of the seconds value register for the seconds field of the RTC. When read, this returns the latest value of the counter. This register and the registers defined in Table 50 and in Table 51 are linked. When the nanoseconds value register is read (see Table 50), the entire RTC is sampled.

Table 52: **Current RTC Seconds Field Value Bits [47:32] (0x1281C)**

Bits	Default Value	Type	Description
15-0	0	RO	Sampled Value of the synchronized RTC Seconds field (bits 47-32).
32-16	0	RO	Reserved

RTC Interrupt Clear Register

Table 53 describes the control register defined for the `interrupt_ptp_timer` signal, the periodic interrupt signal which is raised by the RTC.

Table 53: **RTC Interrupt Clear Register (0x12820)**

Bits	Default Value	Type	Description
0	0	WO	Write ANY value to bit 0 of this register to clear the `interrupt_ptp_timer` Interrupt signal. This bit always returns 0 on read.
31-1	0	RO	Reserved

Phase Adjustment Register

Table 54 describes the Phase Adjustment Register which has units of nanoseconds. This value is added to the synchronized value of the RTC nanoseconds field, and the RTC timing signals are then derived from the result. This phase offset is therefore applied to the `clk8k` signal. As an example, writing the value of the decimal 62500 (half of an 8 kHz clock period) to this register would invert the `clk8k` signal with respect to a value of 0. For this reason, this register can provide fine grained phase alignment of these signals to a 1 ns resolution.

Table 54: **RTC Phase Adjustment Register (0x12824)**

Bits	Default Value	Type	Description
29-0	0	R/W	ns value relating to the phase offset for all RTC derived timing signals (clk8k).
31-30	0	RO	Reserved

Verification

The TEMAC solution 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 TEMAC solution has been designed to directly target the SP601, SP605, ML605 and KC705 boards, enabling the example design to be directly downloaded to the relative board with basic packet generation and loopback functionality. This example design has been verified on all currently supported platforms. No hardware verification has been performed for the Virtex®-7 or Artix™-7 devices; this will be done when boards become available.

Device Utilization

Table 55 through Table 57 provide approximate utilization figures for various core options when a single instance of the core is instantiated in a Virtex-7 device. Other families (Spartan®-6, Virtex-6) have similar utilization figures.

Note: Virtex-6 devices support GMII and MII at 2.5 V only; see the Virtex-6 FPGA Data Sheet: DC and Switching Characteristics for more information. For Virtex-7, Kintex™-7 and Artix-7 devices, it is I/O dependant with HR IO supporting MII/GMII or RGMII at 2.5 V or lower and HP IO only supporting 1.8 V or lower., see the relevant FPGA Data Sheet.

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

Table 55 does not differentiate between 10/100/1000 Mb/s support and 1 Gb/s only support or GMII, MII and RGMII Physical Interfaces. The numbers quoted are for GMII 10/100/1000 Mb/s support; 1 Gb/s only support Slice, lookup table (LUT) and flip-flop (FF) figures will be slightly reduced.

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 55: **10/100/1000 Mb/s and 1 Gb/s Device Utilization**

Core Parameters			Device Resources				
Management Interface	AVB Endpoint	Half- Duplex Support	Slices	LUTs	FFs	LUTRAM	BUFGs
AXI4	No	Yes	800	1400	1700	30	3-4
AXI4	No	No	650	1100	1500	30	3-4
AXI4	Yes	No	1300	2700	3200	150	3-5
None	No	Yes	500	900	1100	30	2
None	No	No	400	600	800	30	2

Additional Features

As well as the core utilization shown in Table 55, 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 56: **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 57: **Frame Filter Utilization**

Core Parameters	Device Resources			
Filters	Slices	LUTs	FFs	LUTRAM
0	20	50	20	30
1	50	100	40	60
each additional filter	30	50	20	30

Performance

Performance in Virtex-6 Lower Power Devices

Ethernet MAC limitations:

- 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.

Performance in Spartan 6 Devices

Ethernet MAC limitation:

- 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. Sufficient system margin and IODELAY tap settings are necessary for correct operation. See Xilinx Answer Record 40028 for more details.

References

1. Virtex-6 FPGA Data Sheet: DC and Switching Characteristics (DS152)
2. Ethernet 1000BASE-X PCS/PMA or SGMII User Guide (UG155)
3. 7 Series FPGAs Configuration User Guide (UG470)
4. 7 Series FPGAs Clocking Resources User Guide (UG472)
5. 7 Series FPGAs Configurable Logic Block User Guide (UG474)
6. Spartan-6 FPGA Data Sheets
7. *IEEE 802.3-2008* specification
8. Reduced Gigabit Media Independent Interface (RGMII), version 2.0
9. Tri-Mode Ethernet MAC User Guide (UG777)
10. AXI Ethernet Data Sheet (DS759)

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

Table 58 shows the bundle offerings.

Table 58: **TEMAC Bundle Offerings**

Part Number	License	IP Cores
EF-DI-TEMAC-SITE	SignOnce IP Site License	10/100/1000 Mb/s, 1Gb/s, 10/100 Mb/s
EF-DI-TEMAC-PROJ	SignOnce IP Project License	10/100/1000 Mb/s, 1Gb/s, 10/100 Mb/s
EF-DI-10-100-EMAC-SITE	SignOnce IP Site License	10/100 Mb/s
EF-DI-EAVB-SITE	SignOnce IP Site License	100/1000 Mb/s Ethernet AVB Endpoint

Two free evaluation licenses are provided: The Simulation Only license is provided with the CORE Generator™ software, and the Full-System Hardware Evaluation license, which lets you test your designs in hardware for a limited period of time, can be downloaded from the TEMAC and Ethernet AVB Endpoint product pages.

For full access to all core functionality, both in simulation and in hardware, you must purchase the the TEMAC IP core and optional Ethernet AVB Endpoint IP core. After purchasing, go to the TEMAC product page and optional Ethernet AVB Endpoint product page for more information on generating the relevant license key for use with the Xilinx® CORE Generator System v13.4.

Contact your local Xilinx sales representative for pricing and availability about Xilinx LogiCORE IP modules and software or see the Xilinx IP Center.

List of Acronyms

Acronym	Definition
AVB	Audio Video Bridging
AXI	Advanced eXtensible Interface
BMCA	Best Master Clock Algorithm
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

Acronym	Definition
PMA	Physical Medium Attachment
PMD	Physical Medium Dependent
PTP	Precise Timing Protocol
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
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

Revision History

Date	Version	Revision
3/1/11	1.1	Initial release for AXI support.
1/18/12	2.0	Added Ethernet AVB Endpoint information for ISE13.4 release.

Notice of Disclaimer