Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

LATTICE SEMICONDUCTOR

LatticeCORE

Tri-Speed Ethernet MAC IP User Guide

April 2015
IPUG51_3.3

Table of Contents

This document provides technical information about the Lattice 10/100/1G Tri-Speed Ethernet Media Access Controller (TSMAC) IP core.

The TSMAC IP core supports the ability to transmit and receive data between a host processor and an Ethernet network. The main function of the Ethernet MAC is to ensure that the Media Access rules specified in the 802.3 IEEE standard are met while transmitting a frame of data over Ethernet. On the receiving side, the Ethernet MAC extracts the different components of a frame and transfers them to higher applications through the FIFO interface.

The TSMAC IP core comes with the following documentation and files:

- Protected netlist/database
- Behavioral RTL simulation model
- Source files for instantiating and evaluating the core

Quick Facts

Table 1-1 gives quick facts about the TSMAC IP core.

Table 1-1. TSMAC IP Core Quick Facts

		TSMAC IP Configuration			
		Across All IP Configurations (Classic, Gigabit, SGMII, MIIM)			
Core Requirements	FPGA Families Supported	ECP5, LatticeECP3, LatticeXP2			
	Minimal Device Needed	LFE5UM-25F-6MG285C	LFE5U-25F-6MG285C	LFE3-17E-6FN256C	LFXP2-5E-5F132C
Resource Utilization	Data Path Width	8			
	LUTs	1400-1900			1500-2000
	sysMEM EBRs	1-2			
	Registers	1100-1400			
Design Tool Support	Lattice Implementation	Lattice Diamond® 3.2			
	Synthesis	Synopsys® Synplify® Pro for Lattice I-2013.09L-SP1-1			
		Mentor Graphics® Precision® RTL			
	Simulation	Aldec® Active-HDL™ 9.3 Lattice Edition			
		Mentor Graphics® ModelSim® SE (Verilog only)			

Features

- Compliant to IEEE 802.3-2005 standard
- Generic 8-bit host interface
- 8-bit wide internal data path
- Generic transmit and receive FIFO interface
- Full-duplex operation in 1G mode
- Full- and half-duplex operation in 10/100 mode
- Transmit and receive statistics vector
- Programmable Inter-Packet Gap (IPG)
- Multicast address filtering
- Selectable MAC operating options
  - Classic TSMAC with G/MII
  - Gigabit MAC with GMII
  - SGMII Easy Connect MAC with GMII, configurable option available on ECP5 and LatticeECP3 devices
- Supports
  - Full-duplex control using PAUSE frames
  - VLAN tagged frames
  - Automatic re-transmission on collision
  - Automatic padding of short frames
  - Multicast and Broadcast frames
  - Optional FCS transmission and reception
  - Optional MII management interface module
  - Jumbo frames of any length

Functional Description

The TSMAC IP core is a fully synchronous machine composed of Transmit and Receive MAC sections that operate independently to support full duplex operation.

The block diagram of the TSMAC IP core is shown in Figure 2-1. The major functional modules are:

- Host Interface
- Receive MAC
- Transmit MAC
- Internal Buffers and FIFO Interfaces
- G/MII
- Management interface module (optional)

In the 1G mode, the 125 MHz system clock is supplied to the Transmit MAC. The system clock is used to clock the GMII interface for data transmission. When receiving data, an external PHY device provides the 125 MHz clock to the GMII receive section. The 125 MHz clock is used to clock the Receive MAC.

In the 10/100 mode, an external PHY device supplies the clock to the Transmit MAC and the Receive MAC.

Configuration Options

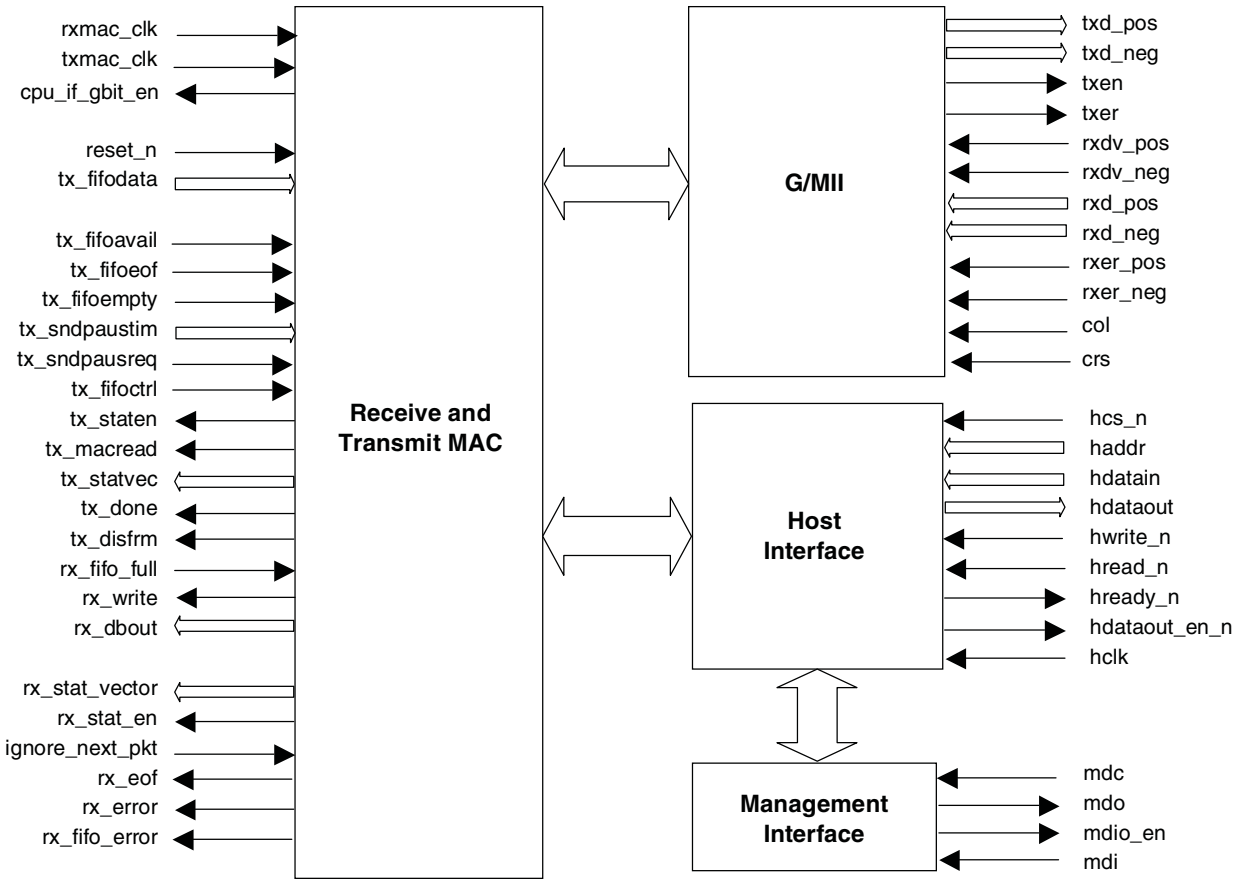
The TSMAC IP core supports three basic configuration options:

- Classic TSMAC with G/MII
- Gigabit MAC with GMII
- SGMII Easy Connect MAC with GMII.

Classic TSMAC Option

When the Classic TSMAC option is selected, the TSMAC IP core can be configured to operate in either the 1G mode (1000Mbps data rate) or the Fast Ethernet mode (10/100 Mbps data rate). A block diagram of the Classic TSMAC IP core option is shown in Figure 2-1. Operation in either 1G mode or Fast Ethernet mode is selected by setting an internal register bit.

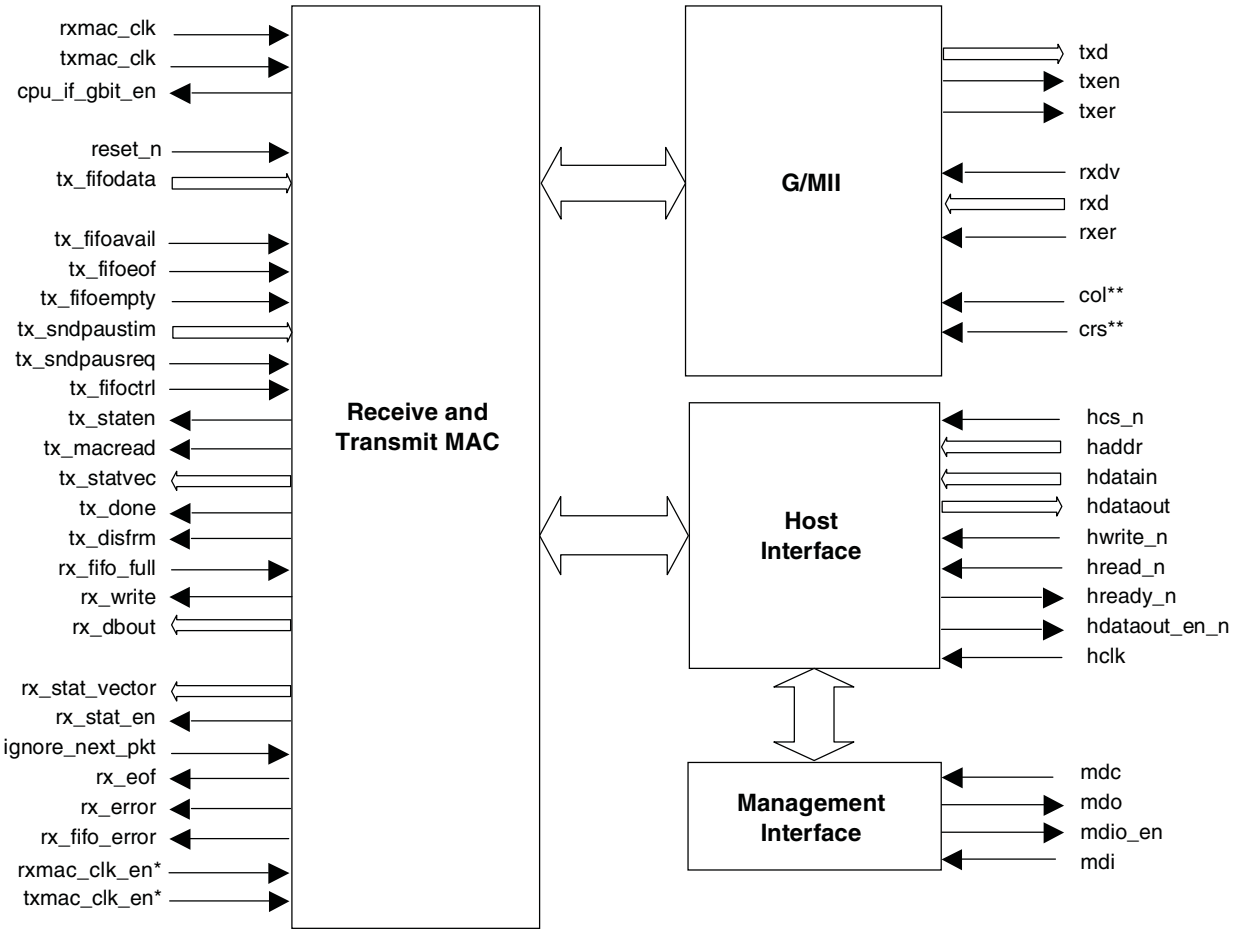
Figure 2-1. Core Block Diagram for the Classic TSMAC IP Core Option



1G or SGMII Configurable Options

For the SGMII Easy Connect configuration option, the TSMAC operates at the Gigabit data rate and uses clock enables provided by the SGMII PCS IP core to work at the three different speeds. For the Gigabit MAC configuration option, the TSMAC always operates at the Gigabit data rate and is effectively configured as a full-duplex Gigabit MAC only. A block diagram of the Gigabit MAC configuration or SGMII Easy Connect configuration is shown in Figure 2-2.

Figure 2-2. Core Block Diagram for the 1G or SGMII Configurable Options



* These inputs are only present for the SGMII Easy Connect option.
** These inputs are not present for the Gigabit MAC option.

Functional Overview

The TSMAC IP core transmits and receives data between a client application and an Ethernet network. The main function of the Ethernet MAC is to ensure that the Media Access rules specified in the 802.3 IEEE standard are met while transmitting and receiving Ethernet frames. Figure 2-3, Figure 2-4, and Figure 2-5 show some of the frame formats of data transmitted and received on the Ethernet network that the TSMAC IP core supports.

On the receiving side, the Ethernet MAC extracts the different components of a frame and transfers them to higher applications through the client FIFO interface.

The data received from the G/MII interface is first buffered until sufficient data is available to be processed by the Receive MAC (Rx MAC). The Preamble and the Start-of-Frame Delimiter (SFD) information are then extracted from the incoming frame to determine the start of a valid frame. The Receive MAC checks the address of the received packet and validates whether the frame can be received before transferring it into the FIFO. Only valid frames are transferred into the FIFO (runts and fragments are discarded). The Rx MAC also provides a statistics vector on a per packet basis that can be used by the application. The TSMAC IP core always calculates CRC to check whether the frame was received error-free.

On the transmit side, the Tx MAC is responsible for controlling access to the physical medium. The Tx MAC reads data from an external client Tx FIFO, formats this data into an Ethernet packet and passes it to the G/MII module.

The Tx MAC reads data from the Tx Client FIFO when the client indicates a packet is available, and the Tx MAC is in its appropriate state. The Tx MAC pre-fixes the Preamble and the Start-of-Frame Delimiter information to the data and appends the Frame Check Sequence at the end of the data. In half-duplex operation, the Tx MAC stores the first 64 bytes of data from the external FIFO in an internal buffer, to be used in re-transmitting data on collisions.

The SGMII Easy Connect configuration option adds pins and logic for seamless connection to the Lattice Gigabit Ethernet PCS IP core.

Figure 2-3. Un-Tagged Ethernet Frame Format

PREAMBLE	SFD	DESTINATION ADDRESS	SOURCE ADDRESS	LENGTH/ TYPE	DATA/PAD	FRAME CHECK SEQUENCE
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	46-1500 bytes	4 bytes

Figure 2-4. VLAN-Tagged Ethernet Frame Format

PREAMBLE	SFD	DESTINATION ADDRESS	SOURCE ADDRESS	VLAN TAG HEADER	LENGTH/ TYPE	DATA/PAD	FRAME CHECK SEQUENCE
7 bytes	1 byte	6 bytes	6 bytes	4 bytes	2 bytes	46-1500 bytes	4 bytes

Figure 2-5. Ethernet Control Pause Frame Format

PREAMBLE	SFD	DESTINATION ADDRESS 01-80-C2-00-00-01	SOURCE ADDRESS	LENGTH/TYPE 88-08	MAC CTL OP_CODE 00-01	OP_CODE PARAMS/RSV	FRAME CHECK SEQUENCE
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	2 bytes	60 bytes	4 bytes

A Tagged frame includes a 4-byte VLAN Tag field, which is located between the Source Address field and the Length/Type field. The VLAN Tag field includes the VLAN Identifier and other control information needed when operating with Virtual Bridged LANs as described in IEEE P802.1Q.

System Block Diagrams

Figure 2-6 shows an FPGA system-level block diagram of how the MAC core is instantiated and used when configured to support the Classic MAC mode of operation. Note that the MAC core needs a certain amount of support logic like PLLs, I/O buffers, and I/O flip flops to meet the IEEE 802.3 I/O specifications (see Application Support for more details).

Figure 2-6. System Block Diagram for the Classic TSMAC IP Core Option

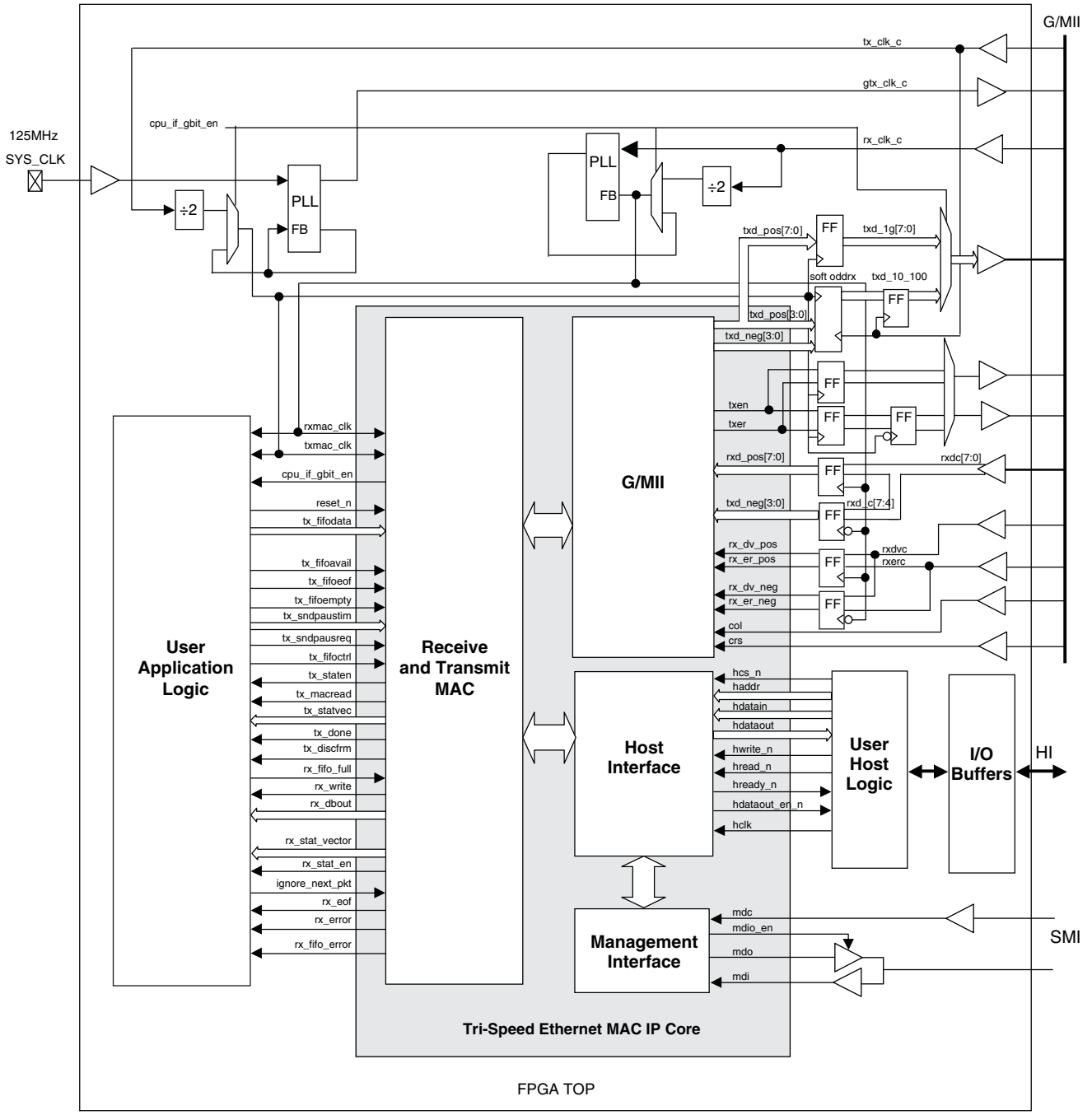


Figure 2-7 shows a FPGA system-level block diagram of how the MAC core is instantiated and used when configured to support the Gigabit MAC mode of operation.

Figure 2-7. System Block Diagram for the Gigabit MAC Configurable Option

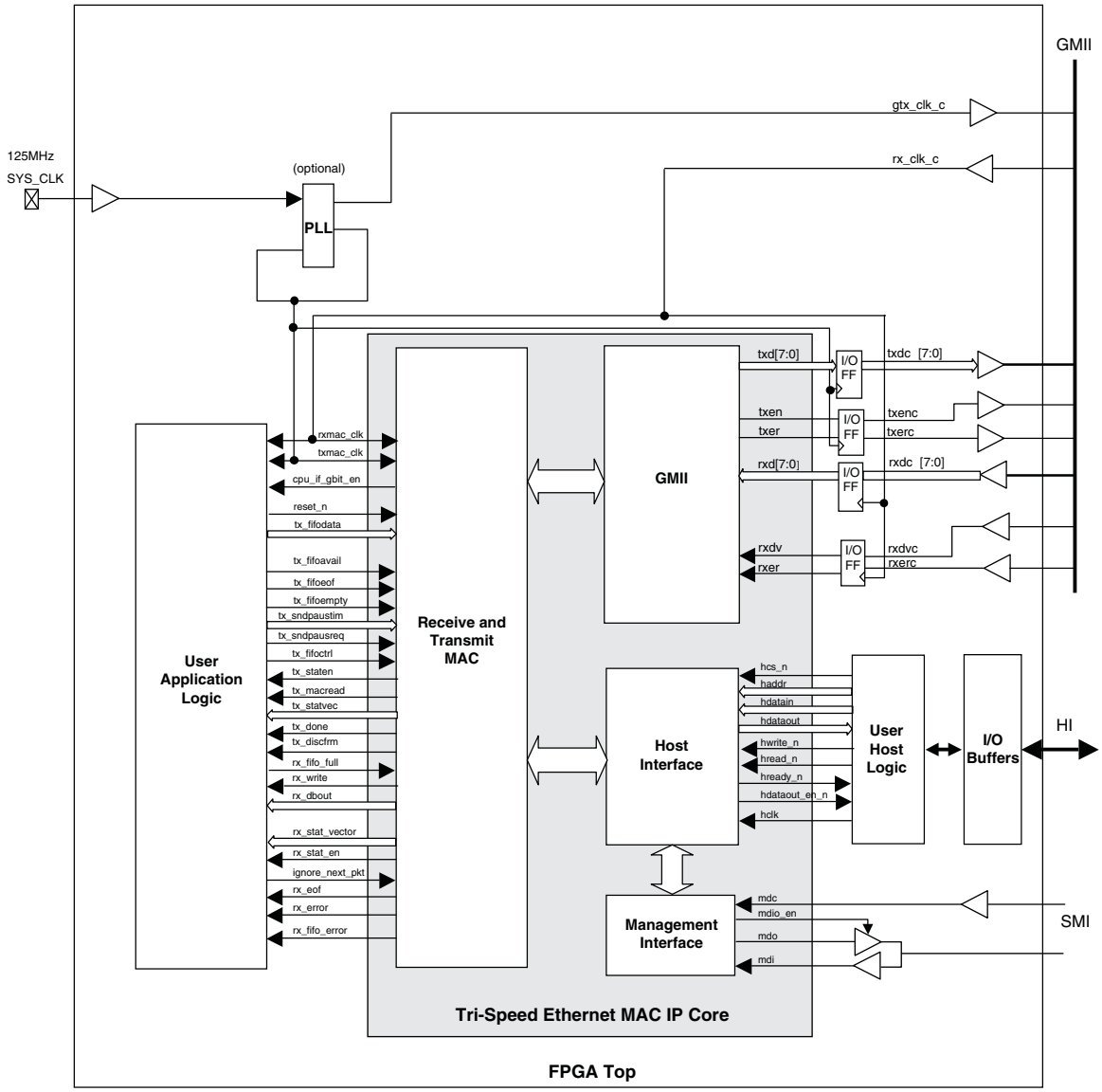
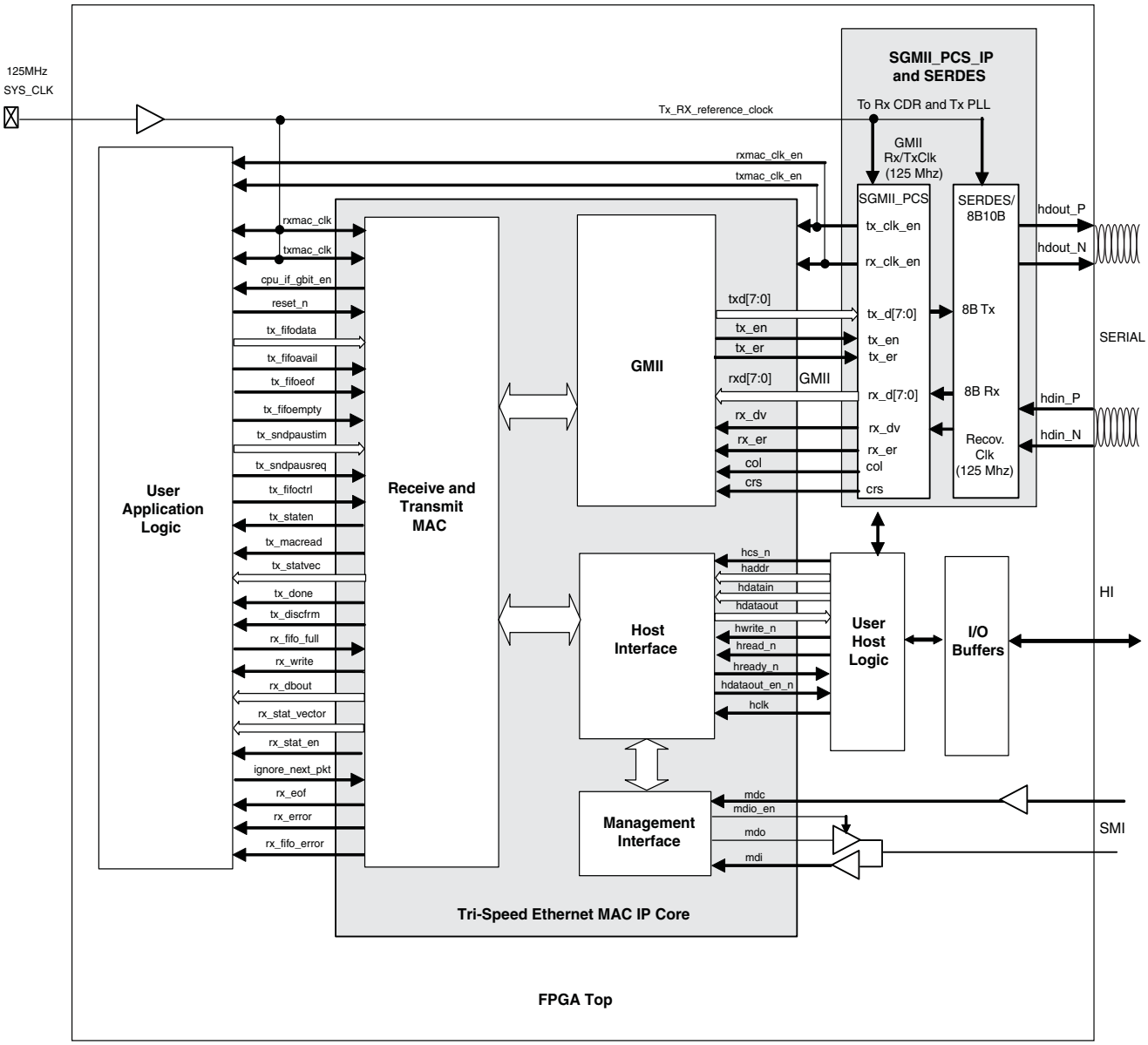


Figure 2-8 shows a FPGA system-level block diagram of how the MAC core is instantiated and used when configured to support the SGMII Easy Connect mode of operation.

Figure 2-8. System Block Diagram for the SGMII Easy Connect Option



Core Signal Descriptions

Table 2-1 lists the I/O signals for the TSMAC IP core.

Table 2-1. TSMAC IP Core Input and Output Signals

Port Name	Type	Active State	Description
Clocks and Reset/Other			
rxmac_clk	Input	N/A	**Receive MAC Application Interface Clock.** This clock is used by the client application and MAC. All outputs driven by the Rx MAC on the client side are synchronous to this clock. This clock's frequency is 125,12.5 or 1.25 MHz depending on the mode 1G/100/10 respectively. *Note: this clock can be viewed as a "byte" clock, since all Rx MAC bytes are aligned with this clock. This clock is derived from the system G/MII rx_clk.* Always 125 MHz in Easy Connect mode.
txmac_clk	Input	N/A	**Transmit MAC Application Interface Clock**. This clock is used by the client application and MAC. All inputs to the Tx MAC on the client side should be synchronous to this clock. This clock's frequency is 125,12.5 or 1.25 MHz depending on the mode 1G/100/10 respectively. *Note: this clock can be viewed as a "byte" clock, since all Tx MAC bytes should be aligned with this clock. This clock is derived from the system sys_clk or tx_clk. (1G or 10/100 respectively).* Always 125 MHz in Easy Connect mode.
hclk	Input	N/A	**Host Clock**. This is the Host Bus clock, and is used to clock the Host Bus interface.
mdc	Input	N/A	**Management Data Clock**. This clock is used only when the Management Interface module is implemented.
reset_n	Input	Low	**Reset**. This is an active low asynchronous signal that resets the internal registers and internal logic. When activated, the I/O signals are driven to their inactive levels.
txmac_clk_en	Input	N/A	**Tx Clock Enable**. This input signal is a clock enable used only in the SGMII Easy Connect option. The SGMII_PCS IP core drives this signal. The clock enable is always high for 1G operation. For 100 Mbps operation the clock enable is asserted high once every ten (125MHz) clocks, and for 10 Mbps operation the clock enable is asserted high once every hundred (125MHz) clocks.
rxmac_clk_en	Input	N/A	**Rx Clock Enable**. This input signal is a clock enable used only in the SGMII Easy Connect option. The SGMII_PCS IP core drives this signal. The clock enable is always high for 1G operation. For 100 Mbps operation the clock enable is asserted high once every ten (125MHz) clocks, and for 10 Mbps operation the clock enable is asserted high once every hundred (125MHz) clocks.
cpu_if_gbit_en	Output	High	**CPU Interface 1G Mode Enabled Indication**. This signal, when high, is an indication from the CPU interface that the 1G mode is enabled. This signal reflects the state of bit 0 of the MAC mode register.
Host Interface			
hcs_n	Input	Low	**Chip Select**. This is an active low signal used to select the core for register Read/Write operations.
haddr[7:0]	Input	N/A	**Address**. This selects one of the internal core registers.
hdatain[7:0]	Input	N/A	**Data Bus Input**. The CPU writes to the internal registers through the data bus.
hwrite_n	Input	Low	**Host Write**. This active low signal is used to write data to the selected register.
hread_n	Input	Low	**Host Read**. This active low signal is used to read data from the selected register.
hready_n	Output	Low	**Ready**. This is an active low signal used to indicate the end of transfer. For write operations, hready_n is asserted after data is accepted (written). For read operations hready_n is asserted after data on the hdataout bus is ready to be driven out.

Table 2-1. TSMAC IP Core Input and Output Signals

Port Name	Type	Active State	Description
hdataout_en_n	Output	Low	**Data Out Enable**. This signal is driven low whenever the TTSMAC IP core outputs valid data onto the hdataout bus. This signal can be used to build a bi-directional data bus.
hdataout[7:0]	Output	N/A	**Data Bus Output**. The CPU reads the internal registers through the data bus.
Transmit MAC Application Interface			
tx_fifodata[7:0]	Input	N/A	**Transmit FIFO Read Data Bus**. The data from the FIFO is presented on this bus.
tx_fifoavail	Input	High	**Transmit FIFO Data Available**. When asserted, this signal indicates that the TxFIFO has data ready for transmission on the G/MII interface. Once this signal is asserted by the client, a short delay later the frame will be transmitted. The client needs to use an appropriate threshold on the client FIFO to indicate that a frame is ready to be sent and use that threshold as the tx_fifo_avail signal.
tx_fifoeof	Input	High	**Transmit FIFO End of Frame**. This signal is asserted along with the last byte of frame data indicating the end of the frame.
tx_fifoempty	Input	High	**Transmit FIFO Empty**. This signal indicates that the TxFIFO is empty. When this signal is asserted and the TSMAC IP core is reading the FIFO, the under-run condition is transferred to the network through the txer signal.
tx_sndpaustim[15:0]	Input	N/A	**PAUSE Frame Timer**. This signal indicates the PAUSE time value that should be sent in the PAUSE frame.
tx_sndpausreq	Input	High	**PAUSE Frame Request**. When asserted, the TSMAC IP core transmits a PAUSE frame. This is also the qualifying signal for the tx_sndpausetim bus.
tx_fifoctrl	Input	N/A	**FIFO Control Frame**. This signal indicates whether the current frame in the Transmit FIFO is a control frame or a data frame. It is qualified by the tx_fifoavail signal. The following values apply: • 1 = Control frame • 0 = Normal frame
tx_staten	Output	High	**Transmit Statistics Vector Enable**. When asserted, the contents of the statistics vector bus tx_statvec are valid.
tx_macread	Output	High	**Transmit FIFO Read**. This is the TSMAC IP core Transmit FIFO read request, asserted by the TSMAC IP core when it intends to read the client FIFO. The MAC core will first assert the tx_macread signal if the client FIFO is not empty (i.e., tx_fifoempty = 0), after which the tx_macread may de-assert (based on MAC processing) or re-assert (based on MAC processing and if tx_fifoempty is still 0). The tx_macread signal should be tied to the client FIFO read pin, and the FIFO empty pin should be tied to the tx_fifoempty of the MAC core.

Table 2-1. TSMAC IP Core Input and Output Signals

Port Name	Type	Active State	Description
tx_statvec[30:0]	Output	N/A	**Transmit Statistics Vector**. This bus includes useful information about the frame that was just transmitted. The corresponding bit locations of this bus are defined as follows: • tx_statvec[0] - UNICAST frame • tx_statvec[1] - Multicast frame • tx_statvec[2] - BROACAST frame • tx_statvec[3] - Bad FCS frame • tx_statvec[4] - JUMBO frame • tx_statvec[5] - FIFO under-run • tx_statvec[6] - PAUSE frame • tx_statvec[7] - VLAN tagged frame • tx_statvec[21:8] - Number of bytes in the transmitted frame • tx_statvec[22] - Deferred transmission • tx_statvec[23] - Excessive deferred transmission • tx_statvec[24] - Late collision • tx_statvec[25] - Excessive collision • tx_statvec[29:26] - Number of early collisions • tx_statvec[30] - FCS generation is disabled and a short frame was transmitted
tx_done	Output	High	**Transmit Done**. This signal is asserted for one clock cycle after transmitting a frame if no errors were present in transmission.
tx_discfrm	Output	High	**Discard Frame**. This signal is asserted at the end of a frame transmit process if the TSMAC IP core detected an error. The possible conditions are: • A FIFO under-run • Late collision (10/100 Mode only) • Excessive Collisions (10/100 Mode only) The user application normally moves the pointer to next frame in these conditions.
Management Interface Signals			
mdi	Input	High	**Management Data Input**. Used to transfer information from the PHY to the management module.
mdo	Output	High	**Management Data Output**. Used to transmit information from the management module to the PHY.
mdio_en	Output	High	**Management Data Out Enable**. Asserted whenever `mdo` is valid. This may be used to implement a bi-directional signal for `mdi` and `mdo`.
G/MII Signals			
txd_pos[7:0][1] txd_neg[3:0][1] txd[7:0][2]	Output	High	**txd_pos[7:4] - Transmit Data Sent to the PHY Chip** - High nibble. In 1G mode, these bits are used as the GMII txd[7:4] bits after they are pipelined outside the core through some I/O flip-flops clocked at 125 MHz (txmac_clk). These bits are not used in the 10/100 mode. See Figure 2-6. **txd_pos[3:0], txd_neg[3:0] - Transmit Data Sent to the PHY Chip - Low nibble.** In both 1G mode and 10/100 mode, both the txd_pos[3:0] and txd_neg[3:0] bits are used to generate the G/MII txd[3:0] bits after they are muxed outside the core through some I/O DDR cells. Note in the 1G mode, txd_pos[3:0] and txd_neg[3:0] will always have the same value, whereas in the 10/100 mode, the txd_pos[3:0] will have the high nibble of the byte transmitted and txd_neg[3:0] will have the low nibble. In the 1G mode, the txmac_clk rate is 125 MHz and in the 10/100 mode, the clock rate is 1.25 MHz and 12.5 MHz respectively. See Figure 2-6. **txd[7:0]** - Transmit Data Sent to the PHY Interface. These GMII Tx data outputs go to the SGMII_PCS IP core (SGMII Easy Connect option) or to the 1G GMII PHY interface (Gigabit interface option). See Figure 2-7 and Figure 2-8.

Table 2-1. TSMAC IP Core Input and Output Signals

Port Name	Type	Active State	Description
txen	Output	High	**Transmit Enable**. Asserted by the TSMAC IP core to indicate the txd bus contains valid frame.
txer	Output	High	**Transmit Error**. Asserted when the TSMAC IP core generates a coding error on the byte currently being transferred.
rxdv_pos[1] rxdv_neg[1] rxdv[2]	Input	High	**Receive Data Valid**. Indicates the data on the rxd bus is valid. Rxdv is used in the SGMII Easy Connect MAC option and the Gigabit MAC option. Rxdv_pos and rxdv_neg are used only in the Classic TSMAC IP core interface option. This signal is pipelined before entering the core with the rx_clk. Note the "_pos" signals are sampled on the rising edge of the rxmac_clk and the "_neg" signals are sampled on the falling edge of the rxmac_clk, before being presented to the MAC core. See Figure 2-6.
rxd_pos[7:0][1] rxd_neg[3:0][1] rxd[7:0][2]	Input	N/A	**Receive Data Bus**. Data is driven by the PHY on these lines, and is valid whenever rxdv is asserted. These signals are pipelined before entering the core with the rxmac_clk. Note the "_pos" signals are sampled on the rising edge of the rxmac_clk and the "_neg" signals are sampled on the falling edge of the rxmac_clk, before being presented to the MAC core. See Figure 2-6. **rxd[7:0] - Receive Data from the PHY Interface**. These GMII Rx data inputs (valid whenever rxdv is asserted) come from the SGMII_PCS IP core (SGMII Easy Connect option) or from the 1G GMII PHY interface (Gigabit MAC option). See Figure 2-7 and Figure 2-8.
rxer_pos[1] rxer_neg[1] rxer[2]	Input	High	**Receive Data Error**. This signal is asserted by the external PHY device when it detects an error during frame reception. This signal is pipelined before entering the core with the rxmac_clk. Note the "_pos" signals are sampled on the rising edge of the rxmac_clk and the "_neg" signals are sampled on the falling edge of the rxmac_clk, before being presented to the MAC core. See Figure 2-6. **rxer - Receive Data Error from the PHY Interface**. This GMII Rx error input comes from the SGMII_PCS IP core (SGMII Easy Connect option) or from the 1G GMII PHY interface (Gigabit MAC option). See Figure 2-7 and Figure 2-8.
col	Input	High	**Collision**. This active-high signal indicates a collision occurred during transmission. This signal is valid for half-duplex operation in Fast Ethernet (10/100) for the Classic and SGMII Easy Connect options only. Otherwise, it is ignored.
crs	Input	High	**Carrier Sense**. This signal, when logic high, indicates the network has activity. Otherwise, it indicates the network is idle. This signal is valid for half-duplex operation in Fast Ethernet (10/100) for the Classic and SGMII Easy Connect options only.
Receive MAC Application Interface			
rx_fifo_full	Input	High	**Receive FIFO Full**. This signal indicates the Rx FIFO is full and cannot accept any more data. This is an error condition and should never happen.
rx_write	Output	High	**Receive FIFO Write**. This signal is asserted by the TSMAC IP core to request a FIFO write.
rx_dbout[7:0]	Output	N/A	**Receive FIFO Data Output**. This bus contains the data that is to be written into the Receive FIFO.

Table 2-1. TSMAC IP Core Input and Output Signals

Port Name	Type	Active State	Description
rx_stat_vector[31:0]	Output	N/A	**Receive Statistics Vector**. This bus indicates the events encountered during frame reception. This bus is qualified by the rx_stat_en signal. The definition of each signal is explained in the Receive MAC section of this user's guide. The corresponding bit locations of this bus are defined as follows: rx_statvec[15:0] - Frame Byte Count rx_statvec[16] - VLAN Tag Detected rx_statvec[17] - Pause Frame rx_statvec[18] - Control Frame tx_statvec[19] - Unsupported Opcode rx_statvec[20] - Dribble Nibble rx_statvec[21] - Broadcast Address rx_statvec[22] - Multicast Address rx_statvec[23] - Receive OK rx_statvec[24] - Length Check Error rx_statvec[25] - CRC Error rx_statvec[26] - Packet Ignored rx_statvec[27] - Carrier Event Previously Seen rx_statvec[28] – Unused rx_statvec[29] - IPG Violation rx_statvec[30] - Short Frame rx_statvec[31] - Long Frame
rx_stat_en	Output	High	**Receive Statistics Vector Enable**. When asserted, this signal indicates that the contents of the rx_stat_vector bus is valid.
ignore_next_pkt	Input	High	**Ignore Next Packet**. This signal is asserted by the host to prevent a Receive FIFO Full condition. The Receive MAC continues dropping packets as long as this signal is asserted. This is an asynchronous signal.
rx_eof	Output	High	**End Of Frame**. Indicates all the data for the current packet has passed on to the FIFO.
rx_error	Output	High	**Receive Packet Error**. When asserted, this signal indicates the packet contains error(s). This signal is qualified with the rx_eof signal. The rx_error signal will be asserted for any of the following three conditions: • The rxer signal on the GMII is asserted by the PHY during frame reception. • There are Rx FCS errors on received frames. • There is a length check error on the received frame.
rx_fifo_error	Output	High	**Receive FIFO Error**. This signal is asserted when the external Rx FIFO is full (rx_fifo_full signal asserted) and the Rx FIFO is being written to by the Rx MAC (rx_write is asserted). When this error signal is asserted the rx_write signal will be de-asserted as long as the rx_fifo_error signal is asserted. The rx_fifo_error signal will be de-asserted when the end of packet (rx_eof) exits the receive FIFO (Note: for this errored condition data will continue to be pulled out of the receive FIFO, even while the rx_write signal has been de-asserted).

1. Classic TSMAC IP core option.
2. Gigabit MAC or SGMII Easy Connect MAC options.

Table 2-2 lists TSMAC IP core system input and output signals.

Table 2-2. TSMAC IP Core System Input and Output Signals

Port Name	Type	Active State	Description
Clocks and Reset			
sys_clk	Input	N/A	**System Clock**. In the 1G mode, the Tx MAC is clocked by this signal. All the input and the output signals of the Tx MAC are synchronous to this clock in the 1G mode. The frequency is always at 125 MHz. Note in the 1G mode the core's txmac_clk is derived from this clock.
tx_clk	Input	N/A	**Transmit Clock**. This clock is used in the 10/100 Mbps mode only. The Tx MAC, Tx MAC application interface and the MII are synchronous to this signal. This clock has a frequency of 2.5/25 MHz for 10/100 Mbps operation respectively. Note in the 10/100 mode tx_clk is divided by two to provide the clock (txmac_clk) to the transmit MAC section. In the 10/100 mode the transmit signals at the GMII interface are always synchronous to tx_clk.
rx_clk	Input	N/A	**Receive Clock**. This clock is an input from the PHY device. In the 1G mode, rx_clk frequency is 125 MHz while in the 10/100 mode, the corresponding rx_clk frequency is 2.5/25 MHz respectively. In the 10/100 mode rx_clk is divided by two to provide the clock (rxmac_clk) to the Receive MAC section. In the 1G mode this clock is provided directly to the Receive MAC section. The receive signals at the GMII interface are always synchronous to rx_clk.
gtx_clk	Output	N/A	**Gigabit Transmit Clock**. This clock is used in the 1G mode only. The transmit signals that are outputs on the GMII interface are synchronous to this clock. This clock has a frequency of 125 MHz. This clock is derived from the sys_clk. See Figure 2-6.

Host Interface

The Host Interface module is a fully synchronous module that runs off the host clock. A number of registers are initialized via the Host interface to ensure that the TSMAC IP core functions as intended. The write operation to an internal register is initiated when the hcs_n and hwrite_n signals are asserted and hread_n signal is deasserted. The address of the targeted register is placed on the haddr bus, while the valid data is placed on the hdatain bus. The contents of the address and data busses should remain unchanged until the TSMAC IP core asserts the hready_n signal. The signals hcs_n, hwrite_n and hread_n must remain unchanged until hready_n is asserted.

A register read is initiated by asserting the hcs_n and hread_n signals, while keeping the hwrite_n signal deasserted. The address of the targeted register is placed on the haddr bus. The TSMAC IP core places the content of the targeted register on the hdataout bus and qualifies it with the assertion of hready_n signal. The haddr bus should not change until the hready_n signal is asserted.

Figure 2-17 shows the timing diagram associated with the host interface write and read operations.

Receive MAC (Rx MAC)

The main function of the Rx MAC is to accept the formatted data from the G/MII interface and pass it to the host application through an external FIFO. In this process, the Rx MAC performs the following functions:

- Detect the start of frame
- Compare the MAC address
- Re-calculate CRC
- Process the control frame and pass it to the flow control module.

The Rx MAC operation is determined by programming the MODE and TX_RX_CTL registers. These register definitions and bit descriptions can be found in Table 2-4. Note that setting the Gbit_en bit in the MODE register to high sets the TSMAC to operate in 1G mode whereas setting the Gbit_en bit to low sets the TSMAC to operate in 10/100 Mode.

Programming the MODE and TX_RX_CTL registers can control the Receive MAC operation. The various events that occur during the reception of a frame are logged into the rx_stat_vector signal and the TX_RX_STS register. At the end of reception, the rx_stat_en signal is asserted to qualify the rx_stat_vector signal. The TSMAC IP core can report a wealth of information such as

- FIFO overflow
- CRC error
- Receive error
- Short frame reception
- Long frame reception
- IPG violation

By default, the entire frame, except the preamble and SFD bytes, is sent to the FIFO via the Rx MAC application interface signals. If the user does not want to receive the FCS, the core can be programmed to strip the FCS field as well as any PAD bytes in the frame and send the rest to the FIFO.

The Rx MAC section operates on the rxmac_clk derived from the rx_clk sourced from the PHY. All the signals on the Receive MAC FIFO interface are synchronous to this clock.

The Rx MAC is disabled while Rx_en is low (Bit_2 of the MODE register) and should only be enabled after the associated registers are properly initialized.

Receiving Frames

The frames received by the Rx MAC are analyzed and the Preamble and SFD bytes are stripped off the frame before it is transferred to an external FIFO. The client data interface between the MAC and the FIFO is eight bits wide.

The default behavior of the MAC is to transfer the unmodified frame after stripping off the Preamble and SFD bytes. This behavior can be changed by setting bit [1] of the TX_RX_CTL register. When bit [1] is set, the Rx MAC strips the Preamble, SFD, FCS bytes and the PAD bytes, if any. Note that the Rx MAC assumes a received frame has PAD bytes if a 64 byte packet is received with its Length/Type field set to a value of less than 46 bytes.

Once the frame is ready to be written into the FIFO, the Rx MAC asserts the rx_write signal, then presents the data on the rx_dbout bus. The rx_write signal is asserted as long as the frame is being written. After transferring the entire frame into the FIFO, the Rx MAC asserts rx_eof indicating the end of the frame. If the frame is received with errors, rx_error is asserted along with rx_eof. If the frame is received with no errors, rx_error remains de-asserted. In either case, a rich set of statistics vectors is presented, containing information about the frame that was received. The statistics vector bus, rx_stat_vector, is qualified by the assertion of rx_stat_en.

If the RxFIFO becomes full, rx_fifo_full is asserted and the frame data is lost. Therefore, the FIFO full condition must be avoided at all times. The rx_fifo_error signal will be asserted along with rx_eof for all frames written into the FIFO while it is full.

The Rx MAC goes to the IDLE state when it is done receiving the frame. This is indicated by bit[10] of the TX_RX_STS register. If the Rx MAC is disabled while it is in the process of receiving a frame, it goes to the IDLE state after it completes the current frame reception.

Address Filtering

The Rx MAC offers several address filtering methods the user can employ to effectively block unwanted frames. It also provides a Promiscuous mode, in which all supported filtering schemes are abandoned and the Rx MAC transfers all the frames irrespective of the address they contain.

By default, the Rx MAC is configured to filter and discard Broadcast frames (i.e. all bits of the received DA == 1) and multicast frames (i.e. bit[0] of the received DA == 1). The MAC can be configured to receive broadcast frames by setting bit [7] of the TX_RX_CTL register.

Multicast frames are received only when bit [4] of the TX_RX_CTL register is set. When set, multicast frames are subject to filtering that is dependent on a 64-bit hash table lookup. The 64-bit hash table is organized as eight, 8-bit registers. The six middle bits of the most significant byte of the CRC calculated for the destination address field of the frame, are used to address one of the 64 bits of the hash table. The three most significant bits of the calculated CRC select one of the eight tables, and the three least significant bits select a bit. The frame is received only if the retrieved bit is set. The IP core registers specifying the hash tables contents are described in Internal Registers. An example of C programming language code that can be used to determine hash table contents based on the multicast addresses to be received is given in Code Listing for Multicast Bit Selection Hash Algorithm in C Language.

All other regular frames are filtered based on the Rx MAC address programmed into the MAC_ADDR_0, MAC_ADDR_1 and MAC_ADDR_2 registers.

Filtering Based on Frame Length

The default minimum Ethernet frame size is 64 bytes. Any frame smaller than 64 bytes could possibly be a collision fragment. By default, the Rx MAC is configured to ignore bytes shorter than 64 bytes. The user can configure the MAC to receive shorter frames by setting bit [8] of the TX_RX_CTL register. Whenever a short frame is received, the appropriate bit is set in the statistics vector, marking it as a Short frame.

The Rx MAC has been designed to receive frames larger than the standard specified maximum as easily as any other frame. This ensures the MAC can work in environments that can generate jumbo frames. However, for statistics purposes, the user can set the maximum length of the frame in the MAX_PKT_SIZE register. When a received frame is larger than the number in this register, bit [31] of the Receive Statistics Vector bus is set, marking it as a Long frame.

Receiving a PAUSE Frame

When the Rx MAC receives a PAUSE frame, the Tx MAC continues with the current transmission, then pauses for the duration indicated in the PAUSE time. During this time, the Tx MAC can transmit Control frames.

Although PAUSE frames may contain the Multicast Address, Multicast filtering rules do not apply to them. If bit [3] of the TX_RX_CTL register is set, the Rx MAC will signal the Tx MAC to stop transmitting for the duration specified in the frame. If this bit is reset, the Rx MAC assumes the Tx MAC does not have the PAUSE capability and/or does not wish to be paused and will not signal it to stop transmitting. If the drop control, bit[6] in the TX_RX_CTL register, is set then the PAUSE frame is received but dropped internal to the MAC and is not transferred to the client FIFO interface. Otherwise, the PAUSE frame is received and transferred to the FIFO.

Statistics Vector

By default, a Statistics Vector is generated for all received frames transferred to the external FIFO. If the user wants the Rx MAC to ignore all incoming frames, then the input signal ignore_next_pkt must be asserted. In this case, a frame that should have been received is ignored and the Rx MAC sets the Packet Ignored bit (bit 26) of the Statistics Vector.

The MAX_PKT_SIZE register is programmed by the user as a threshold for setting the Long Frame bit of the Statistics Vector. This value is used for un-tagged frames only. The Receive MAC will add “4” to the value specified in this register for all VLAN tagged frames when checking against the number of bytes received in the frame. This is because all VLAN tagged frames have an additional four bytes of data.

When a tagged frame is received, the entire VLAN tag field is stored in the VLAN_TAG register. Additionally, every time a statistics vector is generated, some of the bits are written into the corresponding bit locations [9:1] of the TX_RX_STS register. This is done so the user can get this information via the Host interface.

The description of the bits in the Statistics Vector bus is shown in Table 2-3.

Table 2-3. Receive Statistics Vector Descriptions

Bit	Description
31	**Long Frame**. This bit is set when a frame longer than specified in the **MAX_PKT_SIZE** register is received.
30	**Short Frame**. This bit is set when a frame shorter than 64 bytes is received.
29	**IPG Violation**. This bit is set when a frame is received before the IPG timer runs out (96 bit times).
28	**Not Used**. This bit always returns a zero.
27	**Carrier Event Previously Seen**. When asserted, indicates that a carrier event was detected since the last frame.
26	**Packet Ignored**. When set, this bit indicates the incoming packet is to be ignored.
25	**CRC Error**. This bit is set when a frame is received with an error in the CRC field.
24	**Length Check Error**. This bit is set if the number of data bytes in the incoming frame do not match the value in the length field of the frame.
23	**Receive OK**. This bit is set if the frame is received without any error.
22	**Multicast Address**. This bit is set to indicate the received frame contains a Multicast Address.
21	**Broadcast Address**. This bit is set to indicate the received frame contains a Broadcast Address.
20	**Dribble Nibble**. This bit is set when only four bits of the data presented on the RS interface are valid.
19	**Unsupported Opcode**. This bit is set if the received control frame has an unsupported opcode. In this version of the IP, only the opcode for PAUSE frame is supported.
18	**Control Frame**. This bit is set to indicate that a Control frame was received.
17	**PAUSE Frame**. This bit is set when the received Control frame contains a valid PAUSE opcode.
16	**VLAN Tag Detected**. This bit is set when the TSMAC IP core receives a VLAN Tagged frame.
15:0	**Frame Byte Count**. This bus contains the length of the frame that was received. The frame length includes the DA, SA, L/T, TAG, DATA, PAD and FCS fields.

Transmit MAC (Tx MAC)

The Tx MAC is responsible for controlling access to the physical medium. The Tx MAC reads data from an external Tx FIFO when the FIFO is not empty and it detects an active tx_fifoavail. The Tx MAC then formats this data into an Ethernet packet and passes it to the G/MII module.

The Tx MAC is disabled while Tx_en is low (Bit_3 of the MODE register) and should only be enabled after the associated registers are properly initialized. Once enabled, the Tx MAC will continuously monitor the FIFO interface for an indication that frame(s) are ready to be transmitted. In the 1G mode, Tx MAC and the Tx FIFO interface operations are synchronous to txmac_clk derived from sys_clk. In the 10/100 mode, the Tx MAC is clocked by txmac_clk (derived from the tx_clk supplied from the PHY device). The Tx FIFO interface signals are always synchronous to txmac_clk.

In 10/100 mode, the Tx MAC can be configured to operate in the half-duplex or full-duplex mode. This is done by writing to bit[5] of the TX_RX_CTL register. In full-duplex operation, it is possible for the receiver's buffer to fill up rapidly. In such cases, the receiver sends flow control (PAUSE) frames to the transmitter, requesting that it stop transmitting frames. When the receiver is able to free the buffers, the transmitter completes transmitting the current frame and stops for the duration specified in the PAUSE frame.

Transmitting Frames

By default, the Transmit MAC is configured to generate the FCS pattern for the frame to be transmitted. However, this can be prevented by setting bit[2] of the Tx_RX_CTL register. This feature is useful if the frames being presented for transmission already contain the FCS field. When FCS field generation by the MAC is disabled, it is the user's responsibility to ensure that short frames are properly padded before the FCS is generated. If the MAC receives a frame shorter than 64 bytes when FCS generation is disabled, the frame is sent as is and a statistic vector for the condition is generated.

The DA, SA, L/T, and DATA fields are derived from higher applications through the FIFO interface and then encapsulated into an Un-tagged Ethernet frame. This frame is not sent over the network until the network has been idle for a minimum of Inter-Packet Gap (IPG) time. The Frame encapsulation consists of adding the Preamble bits, the Start of Frame Data (SFD) bits and the CRC check sum to the end of the frame (FCS). If padding is not disabled, all short frames are padded with hexadecimal 00.

The input signal tx_fifoeof is asserted along with the last set of data transfer to indicate the end of the frame. The Tx MAC requires a continuous stream of data for the entire frame. There cannot be any bubbles of “no data transfer” within a frame. If the MAC is able to transmit the frame without any errors, the tx_done signal is asserted. Once the transmission has ended, data on the tx_stat_vector bus is presented to the host, including all the statistical information collected in the process of transmitting the frame. Data on this bus is qualified by assertion of the tx_staten signal.

After the Transmit MAC is done transmitting a frame, it waits for more frames from the FIFO interface. During this time, it goes to an idle state that can be detected by reading the TX_RX_STS register. Since the MODE register can be written at any time, the Tx MAC can be disabled while it is actively transmitting a frame. In such cases, the MAC will completely transmit the current frame and then return to the idle state. The control registers should be programmed only after the MAC has returned to the IDLE state.

External Transmit FIFO

The interface between the Tx MAC and the external, client side FIFO is eight bits wide. The bit presented on position 0 is transmitted first and the bit in position 7 is transmitted last. In other words, bit[0] will be transmitted on the txd[0] signal of GMII while the bit[7] will be transmitted on txd[7].

Logic inside the MAC signals if the frame ready for transmission at the head of the FIFO is a Control frame. This is done so the Tx MAC can continue transmission of a Control frame while it is paused.

FIFO Under-flow

If a FIFO underflow occurs, the FIFO logic must assert tx_fifoempty. If at least 64 bytes have been transmitted, the Tx MAC aborts the transmission by asserting tx_er. In addition, the Tx MAC inserts erroneous CRC bits into the packet to guarantee the receiver will detect the error in the packet. If less than 64 bytes have been transmitted when the FIFO underflow occurs, the MAC will pad the remaining bytes before ending the transmission. In either case, the MAC asserts tx_discfrm indicating an error during transmission.

Transmitting PAUSE Frame

Two different methods are used for transmitting a PAUSE frame. In the first method, the application layer forms a PAUSE frame and submits it for transmission via the FIFO. In the other method, the application layer signals the Tx MAC directly to transmit a PAUSE frame. This is accomplished by asserting tx_sndpausreq. In this case the Tx MAC will complete transmission of the current packet and then transmit a PAUSE frame with the PAUSE time value supplied through the tx_sndpaustim bus.

Retries on Collision

When operating in the half-duplex mode, the Transmit MAC has the capability to perform re-transmission of frames that have experienced in-window collision up to the specified maximum. This is possible because the MAC always buffers the first 64 bytes of the frame.

If the MAC has been disabled while it is backing off (soon after a collision), it will only return to the IDLE state after it has successfully transmitted the frame or has exceeded the retry limit.

In the 10/100 mode, the Tx MAC provides the following information:

- Whether the frame deferred before transmission
- The number of times the frame experiences collision before transmission.

This information is sent as a part of the statistics vector. For a frame transmitted without any errors, the statistics vector, qualified by the enable signal, is asserted along with the tx_done signal.

When the frame experiences excessive collision or late collision, the statistics bit for the appropriate condition is set and the tx_discfrm signal is asserted. This indicates an error condition.

Internal Data Buffer and FIFO Interfaces

In the Classic TSMAC IP core and SGMII Easy Connect option, the transmit and receive sections each contain internal FIFO buffers. In the Gigabit MAC option, only the Rx internal FIFO buffer is used. Note that External Transmit and Receive FIFOs (that interface to the MAC client side) are still required to store variable-length normal packets.

On the receive side, the internal FIFO buffer is used to support the dropping of packets less than 64 bytes and to provide additional data buffering for normal packets. The core provides a feature where the user can block all the frames that are shorter than the minimum frame length of 64 bytes in the TSMAC IP core itself (for example collision fragments, runt frames, and such). This prevents these frames from reaching the user's application.

On the transmit side, the internal FIFO buffer stores the first 64 bytes of the frame. This ensures that the TSMAC IP core can re-transmit the frame automatically without help from the application software during an in-window collision. This important feature prevents the propagation of collision information into the application software.

The TSMAC IP core provides two independent interfaces for use with external Transmit and Receive FIFOs. This feature enables the TSMAC IP core to support full duplex operation in either 10/100 or 1G modes.

G/MII Interface

The G/MII module uses the clock supplied by the external PHY. The core implements the standard G/MII interface to connect to the PCS layer.

The module implementing the interface also converts the data to a format usable by the MAC. In the 1G mode, the 8-bit data at the interface is presented to the 8-bit data path of the MAC. In the 10/100 mode, the 4-bit MII data is packed and input to the 8-bit data path of the MAC.

Although not implemented as a separate module, the Reconciliation Sub-layer is implemented as a part of the G/MII interface. This module is responsible for passing the data from one clock domain (TSMAC IP core) to the other G/MII.

(Optional) Media Independent Interface Management Module (MIIM)

The MIIM accesses management information from the PHY device and writes to or reads from the PHY registers. A single MIIM can address up to 32 PHY devices. This module runs off its own clock called mdc. The standard specifies this clock to be at 2.5MHz, but PHY devices can accept a 10MHz mdc clock. Therefore, the TSMAC IP core can have a MIIM that is capable of running at up to 10MHz.

The MIIM read or write operations are specified in the GMII_MNG_CTL register. This register also specifies the addressed PHY and the register within the PHY that needs to be accessed. The Command Finished bit in the GMII_MNG_CTL register is reset as soon as a command to read or write is given. It is set only when the MIIM module completes the operation. While the interface is busy, the GMII_MNG_CTL register cannot be overwritten, and all write operations to the register are ignored. For a write operation, the data to be written is stored in the GMII_MNG_DAT register. For a read operation, the data read from the addressed PHY is stored in this register. The ready bit in the GMII_MNG_CTL is set at the end of the read/write operation.

Internal Registers

The TSMAC IP core internal registers are initialized through the generic Host Interface. These rules apply when accessing the internal registers:

- With the 8-bit Host Interface, the individual bytes of the registers are accessed through their corresponding addresses, with the lower address pointing to the lower byte.
- The reserved bits should be programmed to 0. These bits are invalid, and should be discarded when read.
- All registers except the MODE and GMII Management registers can be written into only when the core is disabled, i.e., MAC is in the IDLE state (Tx_en and Rx_en low in the MODE register). The MODE and GMII Management registers are the only registers that can be written to after the TSMAC IP core is no longer disabled.

Table 2-4 lists the TSMAC IP core registers accessible via the Host Interface. The registers are either Read/Write (R/W) or Read Only (RO) for status reporting purposes. The values of the registers immediately after the Reset Condition is removed from the TSMAC IP core (POR Value in Hexadecimal format) are also given.

Table 2-4. TSMAC IP Core Internal Registers

Register Description	Mnemonic	I/O Address	POR Value
Mode register	MODE	00H - 01H	0000H
Transmit and Receive Control register	TX_RX_CTL	02H - 03H	0000H
Maximum Packet Size register	MAX_PKT_SIZE	04H - 05H	05EEH
Inter-Packet Gap register	IPG_VAL	08H - 09H	000CH
TSMAC IP core Address register 0	MAC_ADDR_0	0AH - 0BH	0000H
TSMAC IP core Address register 1	MAC_ADDR_1	0CH - 0DH	0000H
TSMAC IP core Address register 2	MAC_ADDR_2	0EH - 0FH	0000H
Transmit and Receive Status	TX_RX_STS	12H - 13H	0000H
GMII Management Interface Control register	GMII_MNG_CTL	14H - 15H	0000H
GMII Management Data register	GMII_MNG_DAT	16H - 17H	0000H
VLAN Tag Length/type register	VLAN_TAG	32H - 33H	0000H
Multicast_table_0	MLT_TAB_0	22H - 23H	0000H
Multicast_table_1	MLT_TAB_1	24H - 25H	0000H
Multicast_table_2	MLT_TAB_2	26H - 27H	0000H
Multicast_table_3	MLT_TAB_3	28H - 29H	0000H
Multicast_table_4	MLT_TAB_4	2AH - 2BH	0000H
Multicast_table_5	MLT_TAB_5	2CH - 2DH	0000H
Multicast_table_6	MLT_TAB_6	2EH - 2FH	0000H
Multicast_table_7	MLT_TAB_7	30H - 31H	0000H
Pause_opcode	PAUS_OP	34H - 35H	0080H