PLB 1-Gigabit Ethernet Media Access Controller (MAC) with DMA - PRELIMINARY

DS460 (v1.7.1) August 22, 2003

Product Overview

Introduction

This document provides the design specification for the 1 Gbs Ethernet Media Access Controller (GEMAC) with DMA. The GEMAC described in this document has been designed incorporating the applicable features described in IEEE Std. 802.3-2000. Differences between that specification and the Xilinx GEMAC implementation are highlighted and explained in Specification Exceptions.

The GEMAC Interface design is a soft intellectual property (IP) core designed for implementation in a Virtex™II or Virtex™II Pro FPGA. The GEMAC supports the IEEE Std. 802.3 Gigabit Media Independent Interface (GMII) to industry standard Physical Layer (PHY) devices for full duplex only applications.

For designs in Virtex™II or Virtex™II Pro devices, including the optional Physical Coding Sublayer (PCS) function allows the GEMAC to support the standard Ten Bit Interface (TBI) to external PHY devices. For designs in Virtex™II Pro devices, including the optional Physical Media Attachment (PMA) function with the PCS function allows the GEMAC to take advantage of the built-in Multi-Gigabit Transceivers (MGT) for a greatly reduced signal count SerDes interface to external transceivers. This option greatly reduces routing complexity in the Printed Wiring Board (PWB).

The GEMAC communicates to a processor via a 64-bit IBM Processor Local Bus (PLB) interface, which provides a 1 Gigabit per second full duplex only Ethernet Interface.

The Xilinx GEMAC design allows the customer to tailor the GEMAC to suit their application by setting certain parameters to enable/disable features. Parameterizable features of the design are discussed in GEMAC Design Parameters.

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Virtex™II Pro, Virtex™II	
Version of Core	plb_gemac	v1.00b
Resources Used		
	Min	Max
I/O	387	474
LUTs	2834	5461
FFs	2286	3421
Block RAMs	8	38
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs	None	
Design Tool Requirements		
Xilinx Implementation Tools	5.1i or later	
Verification	N/A	
Simulation	ModelSim SE/EE 5.6e or later	
Synthesis	XST	
Support		
Support provided by Xilinx, Inc.		

The GEMAC is comprised of two, three, or four IP blocks as shown in Figure 1: The IP Interface (IPIF) block is a subset of PLB bus interface features chosen from the full set of

IPIF features to most efficiently couple the second block, the GEMAC core, to the PLB processor bus for this packet based interface. The optional third (PCS) and fourth (PMA) blocks provide flexibility for connection to external Ethernet physical layer devices. This combined entity is referred to as a device. Although there are separate specifications for the IPIF design, this specification addresses the specific implementation required for the GEMAC design

GEMAC Endianess

Please note that the GEMAC is designed as a big endian device (bit 0 is the most significant bit and is shown on the left of a group of bits).

The 8-bit GMII transmit and receive data interface to the external PHY is little endian (bit 7 is the most significant bit and appears on the left of the bus). The MII management interface to the PHY is serial with the most significant bit of a field being transmitted first.

Features

The GEMAC is a soft IP core designed for Xilinx FPGAs and contains the following features:

- 64-bit PLB master and slave interfaces.
- Memory mapped direct I/O interface to registers and FIFOs as well as Simple DMA and Scatter/Gather DMA capabilities for low processor and bus utilization..
- Optional Media Independent Interface Management (MIIM) for access to PHY transceiver registers
- GMII interface to external PHY devices
- Optional PCS function with Ten Bit Interface (TBI) to external PHY devices.
- Option PCS/PMA functions with SerDes interface to external transceiver devices for reduced signal count
- Independent internal 2K, 4K, 8K, 16K, or 32K byte TX and RX FIFOs for holding data for more than one packet (2K byte depth is sufficient for normal 1518 maximum byte packets but 4K byte depth provides better throughput. 16K or 32K byte depth is required for Jumbo frames up to 9K bytes long)
- 16, 32 or 64 entry deep FIFOs for the Transmit Length, Receive Length, and Transmit Status registers to support multiple packet operation
- Filtering of "bad" receive packets to reduce processor bus utilization
- Programmable PHY reset signal
- Auto pad and Frame Check Sequence (FCS) field insertion or pass through on transmit
- Auto pad and FCS field stripping or pass through on receive
- Processes transmission and reception of Pause frames for flow control
- Supports receive and transmit of longer VLAN type frames
- Programmable interframe gap
- Provides interrupts for many error and status conditions
- Optional support of jumbo frames up to 9K bytes in length
- No receive destination address validation. All properly formed packets are accepted.

Ethernet Protocol

Ethernet data is encapsulated in frames as shown in Figure 1 for standard Ethernet, Figure 2 for VLAN Ethernet, Figure 3 for jumbo frames, and Figure 4 for pause/flow control frames[1]. The fields in the frame are transmitted from left to right. The bits within the frame are transmitted from left to right (from least significant bit to most significant bit unless specified otherwise).

Preamble

The preamble field is used for synchronization and must contain seven bytes with the pattern 10101010. The pattern is transmitted from left to right. For transmission, this field is always automatically inserted by the GEMAC and should never appear in the packet data provided to the GEMAC. For reception, this field is always stripped from the packet data.

1. The GEMAC design does not support the Ethernet 8-byte preamble frame type

Start Frame Delimiter

The start frame delimiter field marks the start of the frame and must contain the pattern 10101011. The pattern is transmitted from left to right. The receive data valid signal from the PHY (RX_DV) may go active during the preamble but will be active prior to the start frame delimiter field. For transmission, this field is always automatically inserted by the GEMAC and should never appear in the packet data provided to the GEMAC. For reception, this field is always stripped from the packet data.

Destination Address

The destination address field is 6 bytes in length[1]. The least significant bit of the destination address is used to determine if the address is an individual/unicast (0) or group/multicast (1) address. Multicast addresses are used to group logically related stations. The broadcast address (destination address field is all 1's) is a multicast address that addresses all stations on the LAN. The GEMAC supports transmission and reception of unicast, multicast and broadcast packets.

All properly formed receive packets are accepted regardless of destination address type or value.

Source Address

The source address field is 6 bytes in length[2]. This field is transmitted with the least significant bit first.

Type/Length

The type/length field is 2 bytes in length. When used as a length field, the value in this field represents the number of bytes in the following data field. This value does not include any bytes that may have been inserted in the padding field following the data field. The value of this field determines if it should be interpreted as a length as defined by the IEEE 802.3 standard or a type field as defined by the Ethernet protocol.

The maximum length of a data field is 1,500 bytes for normal frames. Therefore, a value in this field that exceeds 1,500 (05DC hex) would indicates that a frame type rather than a length value is provided in this field. The IEEE 802.3 standard uses the value 1536 (0600 hex) or greater to signal a type field and that is what is used in the GEMAC design. Jumbo frames can have a data field as large as 8982 bytes.

For reception, if the field is a length field and jumbo frames are disabled, the GEMAC will compare the length against the actual data field length and will flag an error if they are different. If the field is a type field or jumbo frames are enabled, the GEMAC will ignore the value and pass it along with the packet data with no further processing. A type/length field value of 8100 hex indicates that the frame is a VLAN frame and a value of 8808 hex indicates a pause MAC control frame.

If the frame is a VLAN type frame, the GEMAC must accept 4 additional bytes which are provided with the received packet data (i.e., the maximum normal frame size is increased from 1518 bytes to 1522 bytes). No additional processing is performed by the GEMAC other than to accept the additional bytes.

The GEMAC does not perform any processing of the type/length field on transmissions. The data provided in the transmit packet is transmitted without any interpretation or validation.

This field is transmitted with the least significant bit first but with the high order byte first. This field is always provided in the packet data for transmissions and is always retained in the receive packet data.

1. The GEMAC design does not support 16-bit destination addresses as defined in the IEEE 802 standard
2. The GEMAC design does not support 16-bit source addresses as defined in the IEEE 802 standard

Data

The data field may vary from 0 to 1500 bytes in length for a normal frame and up to 8982 bytes for a jumbo frame. This field is transmitted with the least significant bit first. This field is always provided in the packet data for transmissions and is always retained in the receive packet data.

Pad

The pad field may vary from 0 to 46 bytes in length. This field is used to insure that the frame length is at least 64 bytes in length (the preamble and SFD fields are not considered part of the frame for this calculation). The values in this field are used in the frame check sequence calculation but are not included in the length field value if it is used.

The length of this field and the data field combined must be at least 46 bytes. If the data field contains 0 bytes, the pad field will be 46 bytes. If the data field is 46 bytes or more, the pad field will have 0 bytes.

For transmission, this field may be inserted automatically by the GEMAC or may be supplied as part of the packet data provided to the GEMAC as indicated by a bit in the GEMAC control register[1].

If GEMAC insertion of padding is enabled in the GEMAC control register, the number of pad bytes to be inserted will determined by the transmit data length register and the FCS and Source address insertion enable bits in the GEMAC control register resulting in the following formula: **PAD (bytes) = 64 - [TXLengthReg + (ENFCS * 4) + (ENSA * 6)]**.

FCS

The FCS field is 4 bytes in length. The value of the FCS field is calculated over the source address, destination address, length/type, data, and pad fields using a 32-bit Cyclic Redundancy Check (CRC) defined as[2]:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + x^0$$

The CRC bits are placed in the FCS field with the x^{31} term in the left most bit of the first byte and the x^0 term is the right most bit of the last byte (i.e., the bits of the CRC are transmitted in the order x^{31}, x^{30},..., x^1, x^0). The GEMAC implementation of the CRC algorithm calculates the CRC value a byte at a time to coincide with the data size exchanged with the external PHY interface for each transmit and receive clock period.

For transmission, this field may be inserted automatically by the GEMAC or may be supplied as part of the packet data provided to the GEMAC as indicated by a bit in the GEMAC control register.

Extension Field (Half duplex only)

The extension field is not used in this full duplex only design and is 0 bytes in length.

1. If the pad field is inserted by the GEMAC, the FCS field will also be calculated and inserted by the GEMAC. This is necessary to insure proper FCS calculation over the pad field. If the pad field is supplied as part of the transmit packet, the FCS may be inserted by the GEMAC or provided as part of the packet to the GEMAC.
2. Reference IEEE Std. 802.3 para. 3.2.8

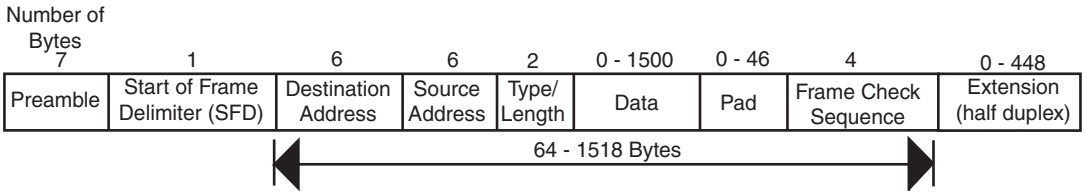

Figure 1: **Ethernet Frame Format**

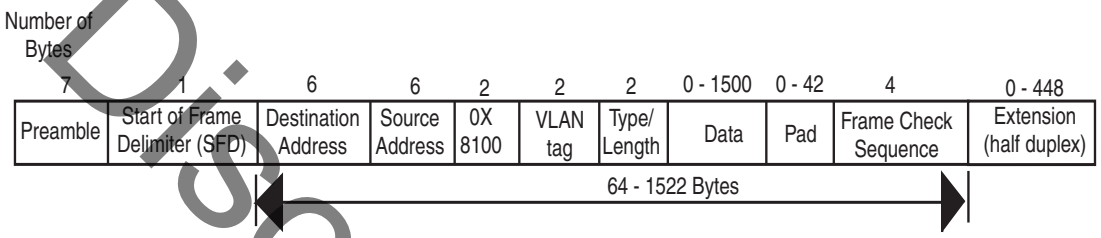

Figure 2: **Ethernet VLAN Frame Format**

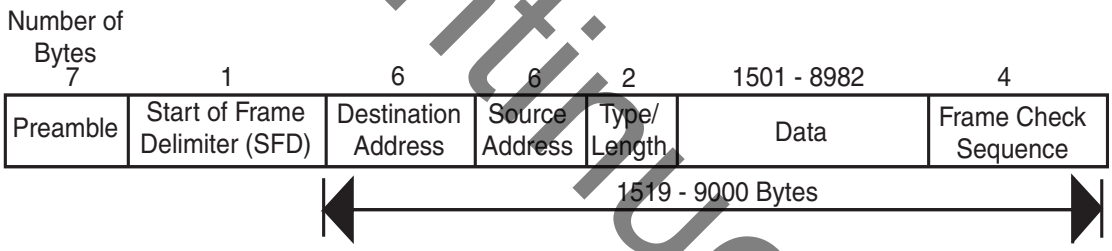

Figure 3: **Ethernet Jumbo Frame Format**

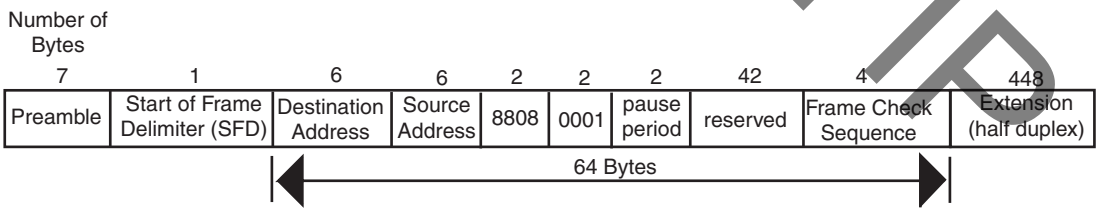

Figure 4: **Ethernet Pause Frame Format**

Interframe Gap[1] and Deferring

Frames are transmitted over the serial interface with an interframe gap which is specified by the IEEE Std. 802.3 to be 96 bit times (0.096 uS for 1 Gbs). This is a minimum value and may be increased with a resulting decrease in throughput (results in a less aggressive approach to gaining access to a shared Ethernet bus). The process for deferring is as follows:

1. Interframe Gap and interframe spacing are used interchangeably and are equivalent.

Full-Duplex

1. The GEMAC does not use the carrier sense signal from the external PHY for full duplex mode since the bus is not shared and only needs to monitor its own transmissions. After the last bit of an GEMAC transmission, the GEMAC starts the interframe gap timer and defers transmissions until it has reached the value represented by the IFGP field of the IFGP register. The minimum gap allowed by the standard is 96 bit times but the GEMAC will allow a minimum gap time of 48 bit times for full duplex.

Flow Control

The GEMAC flow control is designed to Clause 31 of the IEEE 802.3-2000 standard. The GEMAC may be configured to send pause packets/frames and to act upon their reception. These two behaviors can be selected independently in the control register.

Transmitting a Pause Control Frame

To transmit a pause packet, transmission of pause packets must be enabled in the control register. A transmission is initiated when the processor interface writes to the Transmit Pause Packet Register (TPPR).

This will not disrupt any frame transmission in progress but will take priority over any pending frame transmission. This frame will be transmitted even if the transmitter is in the paused state itself.

Receiving a Pause Control Frame

When a pause frame is received by the GEMAC core and reception of pause packets is enabled in the control register, the following checks are made:

- The frame is checked to see whether it is well formed (is a valid Ethernet frame). If not, the frame is dropped.
- If the Destination Address does not match the Pause Control Multicast address (01-80-C2-00-00-01) or the Unicast Address for the GEMAC, the frame is dropped.
- If the Length/Type field does not match the Pause Control Type code (88-08), the frame is dropped.
- If the opcode field contents do not match the PAUSE opcode (00-01), the frame is dropped.

If the frame passes all of these checks, the pause value parameter in the frame is then used to inhibit transmitter operation for the time defined in the IEEE802.3-2000 specification. Since the received pause frame has been acted upon, it is dropped rather than being passed to the receive packet FIFO.

GEMAC Design Parameters

To allow the user to generate an GEMAC that is tailored for their system, certain features are parameterizable in the GEMAC design. This allows the user to have a design that only utilizes the resources required by their system and runs at the best possible performance. The features that are parameterizable in the Xilinx GEMAC design are shown in Table 1.

Table 1: **GEMAC Design Parameters**

Grouping / Number		Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
Top Level	G1	Device Block Id	C_DEV_BLK_ID		1	integer
	G2	BUS clock period in pS	C_PLB_CLK_PERIOD_PS	Requirements as stated in note 1	10000	integer
	G3	Device family	C_FAMILY	virtex2, virtex2p	virtex2p	string
	G4	FIFO depth in bits	C_IPIF_FIFO_DEPTH	262144,131072, 65536, 32768, 16384	262144	integer
	G5	Device base address	C_BASEADDR	See Note 2	None	std logic vector
	G6	Device maximum address	C_HIGHADDR	See Note 2	None	std logic vector
GEMAC Features	G7	Half duplex transmit	C_INCLUDE_HALF_DUPLEX	1 = half and full duplex modules included. Half or full duplex is selectable by software via a bit in the GEMAC control register. 0 = only full duplex modules included	1	integer
	G8	MIIM Interface	C_INCLUDE_MIIM	1 = MIIM Exists 0 = MIIM Non-existent	1	integer
	G9	MIIM Interface Clock Divide	C_MIIM_CLKDVD	00000 to 11111 (indicates the number of times to divide PLB_Clk by 2 to generate an MIIM clock <= 2.5 Mhz) refer to paragraph MII Management Clock	11111	std logic vector
	G10	Statistics counters	C_INCLUDE_STATS	1 = counters Exists 0 = counters Non-existent	1	integer
	G11	GMII interface	C_INCLUDE_GMII	1 = GMII interface Exists 0 = GMII interface Non-existent	1	integer
	G12	GMII & PCS w/ TBI interface	C_INCLUDE_TBI	1 = TBI interface Exists 0 = TBI interface Non-existent	0	integer

Notes:
1. The PLB BUS clock frequency must be in the range of 42 MHz to 125 MHz for 1 Gbs Ethernet operation.
2. No default value will be specified for values to insure that the actual value is set, i.e if the value is not set, a compiler error will be generated. The address range must be at least 3FFF.
3.

Table 1: **GEMAC Design Parameters *(Continued)***

Grouping / Number		Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
	G13	GMII & PCS & PMA w/ SerDes interface	C_INCLUDE_SerDes	1 = SerDes interface Exists 0 = SerDes interface Non-existent	0	integer
PLB/IPIF Interface	G14	Module Identification Read	C_INCLUDE_DEV_MIR	1 = MIR reads Exists 0 = MIR reads Non-existent	1	integer
	G15	Software Reset Function	C_INCLUDE_RESET	1 = software reset Exists 0 = software reset Non-existent	1	integer
	G16	Interrupt device ID encoder	C_INCLUDE_DEV_PENCODER	1 = interrupt device ID encoder Exists 0 = interrupt device ID encoder Non-existent	1	integer
	G17	DMA Type	C_DMA_TYPE	1 = no DMA function is required 2 = simple 2 ch DMA is required 3 = Scatter Gather DMA for packets is required	3	integer
	G18	DMA interrupt coalescing functionality	C_INCLUDE_DMA_INTR_COASLESCE	1	1	integer
	G19	PLB number of masters	C_PLB_NUM_MASTERS	The number of Master Devices connected to the PLB bus	8	integer
	G20	PLB master ID width	C_PLB_MID_WIDTH	The width of the Master ID bus. This is set to roundup(log2 (C_PLB_NUM_MASTERS))	3	integer
	G21	PLB address bus width (in bits)	C_PLB_AWIDTH		32	integer
	G22	PLB data bus width (in bits)	C_PLB_DWIDTH		64	integer

Notes:
1. The PLB BUS clock frequency must be in the range of 42 MHz to 125 MHz for 1 Gbs Ethernet operation.
2. No default value will be specified for values to insure that the actual value is set, i.e if the value is not set, a compiler error will be generated. The address range must be at least 3FFF.
3.

Allowable Parameter Combinations

The GEMAC is a synchronous pipelined design. Due to the pipelined architecture of receive and transmit operations and the interface with the packet FIFOs, the PLB Clock must be in the range of 42 MHz to 125 MHz to allow Ethernet operation at 1 Gbs.

GEMAC I/O Signals

The external I/O signals for the GEMAC are listed in Table 2.

Table 2: **GEMAC I/O Signals**

Grouping		Signal Name	Interface	I/O	Initial State	Description
GEMAC GMII and TBI PHY interface system Clock signal	P1	gtx_clk	System	I		125 Mhz. PHY transmit clocks are derived from this clock.
GEMAC SerDes transceiver interface system clock signals	P2	refclk	System	I		High quality reference clock for Multi-Gigabit transceivers (62.5 Mhz). See Rocket I/O MGT datasheet.
	P3	refclk2	System	I		Alternative high quality reference clock for Multi-Gigabit transceivers (62.5 Mhz). See Rocket I/O MGT datasheet.
	P4	brefclk	System	I		Alternative high quality reference clock for Multi-Gigabit transceivers (62.5 Mhz). This optionally replaces refclk. See Rocket I/O MGT datasheet.
	P5	brefclk2	System	I		Alternative high quality reference clock for Multi-Gigabit transceivers (62.5 Mhz). This optionally replaces refclk2. See Rocket I/O MGT datasheet.

Table 2: **GEMAC I/O Signals *(Continued)***

Grouping		Signal Name	Interface	I/O	Initial State	Description
	P6	refclksel	System	I		Selects between either (b)refclk or (b)refclk2 as the input clock source to the MGT. See Rocket I/O MGT datasheet.
	P7	userclk	System	I		62.5 Mhz. This is connected to the txusrclk and rxusrclk ports of the Rocket I/O MGT. This may be derived from refclk using a DCM as illustrated in Figure 24
	P8	userclk2	System	I		125 Mhz. This is connected to the txusrclk2 and rxusrclk2 ports of the Rocket I/O MGT. This may be derived from refclk using a DCM as illustrated in Figure 24
	P9	DCM_locked	System	I		If a DCM is used to derive usrclk and usrclk2 as illustrated in Figure 24, the locked port of the DCM must be connected to this input. The GEMAC will hold its Rocket I/O MGT in reset until DCM_locked is drive to logic 1.
GEMAC GMII PHY interface signals	P10	phy_rx_data(7:0)	System	I		Ethernet receive data for GMII interface.
	P11	phy_tx_data(7:0)	System	O	00000000	Ethernet transmit data for GMII interface.
	P12	phy_rx_dv	System	I		Ethernet receive data valid indicator for GMII interface.
	P13	phy_rx_er	System	I		Ethernet receive error indicator for GMII interface.

Table 2: **GEMAC I/O Signals *(Continued)***

Grouping		Signal Name	Interface	I/O	Initial State	Description
	P14	phy_tx_en	System	O	0	Ethernet transmit enable for GMII interface.
	P15	phy_tx_er	System	O	0	Ethernet transmit error enable for GMII interface.
	P16	phy_tx_clk	System	O		Ethernet transmit clock for GMII interface.
	P17	phy_rx_clk	System	I		Ethernet receive clock for GMII interface.
	P18	phy_crs	System	I		Ethernet carrier sense indicator for GMII interface.
	P19	phy_col	System	I		Ethernet collision indicator for GMII interface.
	P20	phy_rst_n	System	O	1	Ethernet PHY reset for GMII interface.
GEMAC w/ PCS TBI PHY interface signals	P21	tx_code_group(9:0)	System	O		10-bit transmit data to PMA sublayer
	P22	PMA_tx_clk	System	O		125 Mhz transmit clock to PMA sublayer
	P23	loc_ref	System	O		Causes the PMA sublayer clock recovery unit to lock to PMA_tx_clk. This signal is currently grounded
	P24	ewrap	System	O		When high this instructs the PMA sublayer to electrically loop the transmitter data to the receiver
	P25	rx_code_group0(9:0)	System	I		10-bit receive data from PMA sublayer synchronous to PMA_rx_clk0
	P26	rx_code_group1(9:0)	System	I		10-bit receive data from PMA sublayer synchronous to PMA_rx_clk1

Table 2: **GEMAC I/O Signals *(Continued)***

Grouping		Signal Name	Interface	I/O	Initial State	Description
	P27	PMA_rx_clk0	System	I		62.5 Mhz receive clock signal from PMA sublayer
	P28	PMA_rx_clk1	System	I		62.5 Mhz receive clock signal from PMA sublayer. This is 180 degrees out of phase with PMA_rx_clk0
	P29	en_cdet	System	O		Enables the PMA sublayer to perform comma realignment. This is driven from the PCS sublayer state machine during the "loss-of-sync" state.
GEMAC w/ PCS TBI PHY interface signals or w/ PCS & PMA SerDes transceiver interface signals	P30	PHYad(4:0)	System	I		PHY address of MDIO register set for the PCS sublayer
	P31	signal_detect	System	I		Direct from the PMD sublayer indicating the presence of light detected at the optical receiver. If driven high, this indicates that the optical receiver has detected light. This signal may be permanently tied high to enable operation of the core.
GEMAC w/ PCS & PMA SerDes transceiver interface signals	P32	txp	System	O		one of differential pair for serial transmission from PMA to PMD sublayer. The clock is embedded in the data stream

Table 2: **GEMAC I/O Signals *(Continued)***

Grouping		Signal Name	Interface	I/O	Initial State	Description
	P33	txn	System	O		one of differential pair for serial transmission from PMA to PMD sublayer. The clock is embedded in the data stream
	P34	rxp	System	I		one of differential pair for serial reception from PMD to PMA sublayer. The clock is extracted from the data stream
	P35	rxn	System	I		one of differential pair for serial reception from PMD to PMA sublayer. The clock is extracted from the data stream
GEMAC MIIM Signals	P36	phy_mii_clk_I	System	O	0	GMII management interface clock Input from 3-state output buffer (not used)
	P37	phy_mii_clk_O	System	O	0	GMII management interface clock output to 3-state output buffer
	P38	phy_mii_clk_T	System	O	0	GMII management interface clock enable output to 3-state output buffer
	P39	phy_mii_data_I	System	I		GMII management interface data input from 3-state I/O buffer
	P40	phy_mii_data_O	System	O	0	GMII management interface data output to 3-state I/O buffer
	P41	phy_mii_data_T	System	O	0	GMII management interface data enable output to 3-state I/O buffer

Table 2: **GEMAC I/O Signals *(Continued)***

Grouping		Signal Name	Interface	I/O	Initial State	Description
PLB System Signals (use by masters and slaves)	P42	PLB_Clk	System	I		System clock
	P43	PLB_Rst	System	I		System Reset (active high)
	P44	IP2INTC_Irpt	System	O	0	System Interrupt
	P45	Freeze	System	I		System Freeze Input
PLB Arbiter Signals (used by masters and slaves)	P46	PLB_ABus(0:C_PLB_AWIDTH-1)	IPIF	I		PLB address bus
	P47	PLB_BE(0:(C_PLB_DWIDTH/8)-1)	IPIF	I		PLB byte enables
	P48	PLB_wrDBus(0:C_PLB_DWIDTH-1)	IPIF	I		PLB write data bus
	P49	PLB_RNW	IPIF	I		PLB Read not Write
	P50	PLB_PAValid	IPIF	I		PLB primary address valid indicator
	P51	PLB_SAValid	IPIF	I		PLB secondary address valid indicator
	P52	PLB_rdPrim	IPIF	I		PLB secondary to primary read request indicator
	P53	PLB_wrPrim	IPIF	I		PLB secondary to primary write request indicator
	P54	PLB_masterID(0:C_PLB_MID_WIDTH -1)	IPIF	I		PLB current master indentifier
	P55	PLB_abort	IPIF	I		PLB abort bus request indicator
	P56	PLB_buslock	IPIF	I		PLB bus lock
	P57	PLB_MSize(0:1)	IPIF	I		PLB master data bus size
	P58	PLB_size(0:3)	IPIF	I		PLB transfer size
	P59	PLB_type(0:2)	IPIF	I		PLB transfer type
	P60	PLB_compress	IPIF	I		PLB compressed data transfer indicator
	P61	PLB_guarded	IPIF	I		PLB guarded transfer indicator

Table 2: **GEMAC I/O Signals *(Continued)***

Grouping		Signal Name	Interface	I/O	Initial State	Description
	P62	PLB_ordered	IPIF	I		PLB synchronize transfer indicator
	P63	PLB_lockErr	IPIF	I		PLB lock error indicator
	P64	PLB_wrBurst	IPIF	I		PLB burst write transfer indicator
	P65	PLB_rdBurst	IPIF	I		PLB burst read transfer indicator
	P66	PLB_pendReq	IPIF	I		PLB pending bus request indicator
	P67	PLB_pendPri(0:1)	IPIF	I		PLB pending request priority
	P68	PLB_reqPri(0:1)	IPIF	I		PLB current request priority
PLB Master Signals	P69	PLB_MSSize(0:1)	IPIF	I		PLB slave data bus size
	P70	M_request	IPIF	O	0	Master bus request
	P71	M_priority(0:1)	IPIF	O	0	Master bus request priority
	P72	M_buslock	IPIF	O	0	Master bus lock
	P73	M_RNW	IPIF	O	0	Master Read not Write
	P74	M_BE(0:(C_PLB_DWIDTH/8)-1)	IPIF	O	0	Master byte enables
	P75	M_MSize(0:1)	IPIF	O	0	Master data bus size
	P76	M_size(0:3)	IPIF	O	0	Master transfer size
	P77	M_type(0:2)	IPIF	O	0	Master transfer type
	P78	M_compress	IPIF	O	0	Master compressed data transfer indicator
	P79	M_guarded	IPIF	O	0	Master guarded transfer indicator
	P80	M_ordered	IPIF	O	0	Master synchronize transfer indicator
	P81	M_lockErr	IPIF	O	0	Master lock error indicator
	P82	M_abort	IPIF	O	0	Master abort bus request indicator
	P83	M_abus(0:C_PLB_AWIDTH-1)	IPIF	O	0	Master address bus

Table 2: **GEMAC I/O Signals *(Continued)***

Grouping		Signal Name	Interface	I/O	Initial State	Description
	P84	M_wrDBus(0:C_PLB_DWIDTH-1)	IPIF	O	0	Master write data bus
	P85	M_wrBurst	IPIF	O	0	Master burst write transfer indicator
	P86	M_rdBurst	IPIF	O	0	Master burst read transfer indicator
	P87	PLB_MAddrAck	IPIF	I		PLB master address acknowledge
	P88	PLB_MRearbitrate	IPIF	I		PLB master bus rearbitrate indicator
	P89	PLB_MBusy	IPIF	I		PLB master slave busy indicator
	P90	PLB_MErr	IPIF	I		PLB master slave error indicator
	P91	PLB_MWrDAck	IPIF	I		PLB master write data acknowledge
	P92	PLB_RdDBus(0:C_PLB_DWIDTH-1)	IPIF	I		PLB master read data bus
	P93	PLB_MRdWdAddr(0:3)	IPIF	I		PLB master read word address
	P94	PLB_MRdDAck	IPIF	I		PLB master read data acknowledge
	P95	PLB_MRdBTerm	IPIF	I		PLB master terminate read burst indicator
	P96	PLB_MWrBTerm	IPIF	I		PLB master terminate write burst indicator
PLB Slave Signals	P97	Sl_rdDBus(0:C_PLB_DWIDTH-1)	IPIF	O	0	Slave read data bus
	P98	Sl_addrAck	IPIF	O	0	Slave address acknowledge
	P99	Sl_SSize(0:1)	IPIF	O	0	Slave data bus size
	P100	Sl_wait	IPIF	O	0	Slave wait indicator
	P101	Sl_rearbitrate	IPIF	O	0	Slave rearbitrate bus indicator
	P102	Sl_wrDAck	IPIF	O	0	Slave write data acknowledge
	P103	Sl_wrComp	IPIF	O	0	Slave write transfer complete indicator
	P104	Sl_wrBTerm	IPIF	O	0	Slave terminate write burst transfer

Table 2: **GEMAC I/O Signals *(Continued)***

Grouping	Signal Name	Interface	I/O	Initial State	Description
P105	Sl_rdWdAddr(0:3)	IPIF	O	0	Slave read word address
P106	Sl_rdDAck	IPIF	O	0	Slave read data acknowledge
P107	Sl_rdComp	IPIF	O	0	Slave read transfer complete indicator
P108	Sl_rdBTerm	IPIF	O	0	Slave terminate read burst transfer
P109	Sl_MBusy(0:C_PLB_NUM_MASTERS-1)	IPIF	O	0	Slave busy indicator
P110	Sl_MErr(0:C_PLB_NUM_MASTERS-1)	IPIF	O	0	Slave error indicator

GEMAC Port Dependencies

The width of some of the GEMAC signals depend on parameters selected in the design. The dependencies between the GEMAC design parameters and I/O signals are shown in Table 3.

Table 3: **GEMAC Parameter Port Dependencies**

		Name	Affects	Depends	Relationship Description
Design Parameters	G22	C_PLB_DWIDTH	P47,P48, P74,P84, P92,P97		Specifies the Data Bus width
	G21	C_PLB_AWIDTH	P46,P83		Specifies the Address Bus width
	G19	C_PLB_NUM_MASTERS	P109,P110		Specifies the number of masters on the PLB bus
	G20	C_PLB_MID_WIDTH	P54		Specifies the Master ID bus width
	G17	C_DMA_TYPE	G18		Specifies if DMA is present and which type
	G18	C_DMA_INTR_COASLESCE		G17	Not used if scatter gather DMA not present (G17 is 2)
I/O Signals	P47	PLB_BE(0:(C_PLB_DWIDTH/8)-1)		G22	Width varies with the size of the Data bus.
	P46	PLB_ABus(0:C_PLB_AWIDTH-1)		G21	Width varies with the size of the Address bus.
	P48	PLB_wrDBus(0:C_PLB_DWIDTH-1)		G22	Width varies with the size of the Data bus.
	P74	M_BE(0:(C_OPB_AWIDTH/8)-1)		G22	Width varies with the size of the Data bus.

Table 3: **GEMAC Parameter Port Dependencies *(Continued)***

		Name	Affects	Depends	Relationship Description
	P83	M_ABus(0:C_PLB_AWIDTH-1)		G21	Width varies with the size of the Address bus.
	P84	M_wrDBus(0:C_PLB_DWIDTH-1)		G22	Width varies with the size of the Data bus.
	P92	PLB_RdDBus(0:C_PLB_DWIDTH-1)		G22	Width varies with the size of the Data bus.
	P97	Sl_RdDBus(0:C_PLB_DWIDTH-1)		G22	Width varies with the size of the Data bus.
	P109	Sl_MBusy(0:C_PLB_NUM_MASTERS-1)		G19	Width varies with the number of masters on the PLB bus
	P110	Sl_MErr(0:C_PLB_NUM_MASTERS-1)		G19	Width varies with the number of masters on the PLB bus
	P54	PLB_masterID(0:C_PLB_MID_WIDTH-1)		G20	Width varies with the number of masters on the PLB bus

GEMAC Interrupt Interface

The interrupt signals generated by the GEMAC are managed by the Interrupt Source Controller in the GEMAC IPIF module. This interface provides many of the features commonly provided for interrupt handling. Please refer to the PLB Device Interrupt Architecture specification listed in *Reference Documents*.

Interrupt (data bus bit 31) -- Transmit complete interrupt

Indicates that at least one transmit has completed and that the transmit status word is available.

Interrupt (data bus bit 30) -- Receive complete interrupt

Indicates that at least one successful receive has completed and that the receive status word packet data and packet data length is available. This signal is not set for unsuccessful receives.

Interrupt (data bus bit 29) -- Reserved

Interrupt (data bus bit 28) -- Receive Packet Rejected

Interrupt (data bus bit 27) -- Transmit Status FIFO Empty interrupt

This reflects the status of the transmit status FIFO empty flag. It may be used to indicate that the status words for all completed transmissions have been processed. Any other transmit packets already provided to the GEMAC are either queued for transmit or are currently being transmitted but have not yet completed. This active high signal remains active as long as the condition persists.

Interrupt (data bus bit 26) --Receive Length FIFO Empty interrupt

This reflects the status of the receive length FIFO empty flag. It may be used to indicate that the packet lengths for all successfully completed receives have been processed. The status of this FIFO should always track the status of the receive status FIFO. This active high signal remains active as long as the condition persists.

Interrupt (data bus bit 25) -- Transmit Length FIFO Full interrupt

This reflects the status of the transmit length FIFO full flag. It may be used to pause queueing of transmit packets until some of the queued packets have been processed by the GEMAC. This active high signal remains active as long as the condition persists.

Interrupt (data bus bit 24) -- Receive Length FIFO Overrun interrupt

Indicates that the receive length FIFO became full during the reception of a packet and data was lost. The GEMAC will remove the corresponding packet from the receive data FIFO and no receive status will be stored but to insure that more data is not lost, receive packets stored in the FIFO should be processed to free up more locations. Once set, this bit can only be cleared with a reset.

Interrupt (data bus bit 23) -- Receive Length FIFO Underrun interrupt

Indicates that an attempt was made to read the receive length FIFO when it was empty and that the data received is not valid. Once set, this bit can only be cleared with a reset.

Interrupt (data bus bit 22) -- Transmit Status FIFO Overrun interrupt

Indicates that the Transmit status FIFO became full following the transmission of a packet and data was lost. Care must be taken under these conditions to ensure that the transmit status words do not become out of sync with the originating packet information. To insure that more data is not lost, transmit status words stored in the FIFO should be processed to free up more locations. Once set, this bit can only be cleared with a reset.

Interrupt (data bus bit 21) -- Transmit Status FIFO underrun interrupt

Indicates that an attempt was made to read the transmit status FIFO when it was empty and that the data received is not valid. Once set, this bit can only be cleared with a reset.

Interrupt (data bus bit 20) -- Transmit Length FIFO Overrun interrupt

Indicates that more transmit packets were written to the GEMAC transmit queue than the transmit length FIFO could store and data was lost. This is non-recoverable condition since some or all of the packet data may have been stored in the transmit data FIFO and it can not be removed.

Since there is not a transmit length entry for that packet, the transmit length and data FIFOs are no longer synchronized. This condition should be corrected by forcing an immediate reset of the transmit data FIFO and the transmit path in the GEMAC. Once set, this bit can only be cleared with a reset.

Interrupt (data bus bit 19) -- Transmit Length FIFO Underrun interrupt

Indicates that the GEMAC attempted to remove an entry from the transmit length FIFO following the completion of a transmission and there were no entries in the FIFO. This should never be possible and represents a serious error. This condition should be corrected by forcing an immediate reset of the transmit data FIFO and the transmit path in the GEMAC. condition. Once set, this bit can only be cleared with a reset.

Interrupt (data bus bit 18) -- Reserved

Interrupt (data bus bit 17) -- Reserved

Interrupt (data bus bit 16) -- Reserved

Interrupt (data bus bit 15) -- Reserved

Interrupt (data bus bit 14) -- Receive Max Length Error

Interrupt (data bus bit 13) -- Receive Frame Length Error

Interrupt (data bus bit 12) -- Receive FCS Error

Interrupt (data bus bit 11) -- Receive Slot Length Error

Interrupt (data bus bit 10) -- Receive CRC Engine Error

GEMAC Register & RAM Definition

GEMAC IPIF Registers

The GEMAC design contains registers in each of the two modules (IPIF and GEMAC core). All of the registers in both modules are 32 bits wide or less. As a result, 32 bit addressing will be used for register accesses.

The registers in Table 4 are contained in the IPIF module and are included for completeness of this specification. Detailed descriptions of these registers are provided in the IPIF specifications listed in *Reference Documents*.

The registers in Table 5 are contained in the GEMAC core module and are described in detail in this specification. The addresses for all registers are based on a parameter which is the base address for the entire GEMAC module. The address of each register is then calculated by an offset to the base address.

Table 4: **GEMAC IPIF Registers**

Register Name	PLB ADDRESS	Access
Transmit DMA & Scatter Gather Reset Register	C_DEV_BASEADDR + 0x2300	Write
Transmit DMA & Scatter Gather Module Identification Register	C_DEV_BASEADDR + 0x2300	Read
Transmit DMA & Scatter Gather Control Register	C_DEV_BASEADDR + 0x2304	Read/Write
Transmit DMA & Scatter Gather source address	C_DEV_BASEADDR + 0x2308	Read/Write
Transmit DMA & Scatter Gather destination address	C_DEV_BASEADDR + 0x230C	Read/Write
Transmit DMA & Scatter Gather start/length	C_DEV_BASEADDR + 0x2310	Read/Write
Transmit DMA & Scatter Gather Status Register	C_DEV_BASEADDR + 0x2314	Read
Transmit DMA & Scatter Gather Buffer Descriptor Address	C_DEV_BASEADDR + 0x2318	Read/Write
Transmit DMA Software Control Register	C_DEV_BASEADDR + 0x231C	Read/Write
Transmit DMA & Scatter Gather Unserviced Packet Count	C_DEV_BASEADDR + 0x2320	Read/Write
Transmit DMA & Scatter Gather Packet Count Threshold	C_DEV_BASEADDR + 0x2324	Read/Write
Transmit DMA & Scatter Gather Packet Wait Bound	C_DEV_BASEADDR + 0x2328	Read/Write
Transmit DMA & Scatter Gather Interrupt Status Register	C_DEV_BASEADDR + 0x232C	Read/toggle on Write
Transmit DMA & Scatter Gather Interrupt Enable Register	C_DEV_BASEADDR + 0x2330	Read/Write

Table 4: **GEMAC IPIF Registers *(Continued)***

Register Name	PLB ADDRESS	Access
Receive DMA & Scatter Gather Reset Register	C_DEV_BASEADDR + 0x2340	Write
Receive DMA & Scatter Gather Module Identification Register	C_DEV_BASEADDR + 0x2340	Read
Receive DMA & Scatter Gather Control Register	C_DEV_BASEADDR + 0x2344	Read/Write
Receive DMA & Scatter Gather source address	C_DEV_BASEADDR + 0x2348	Read/Write
Receive DMA & Scatter Gather destination address	C_DEV_BASEADDR + 0x234C	Read/Write
Receive DMA & Scatter Gather start/length	C_DEV_BASEADDR + 0x2350	Read/Write
Receive DMA & Scatter Gather Status Register	C_DEV_BASEADDR + 0x2354	Read
Receive DMA & Scatter Gather Buffer Descriptor Address	C_DEV_BASEADDR + 0x2358	Read/Write
Receive DMA Software Control Register	C_DEV_BASEADDR + 0x235C	Read/Write
Receive DMA & Scatter Gather Unservice Packet Count	C_DEV_BASEADDR + 0x2360	Read/Write
Receive DMA & Scatter Gather Packet Count Threshold	C_DEV_BASEADDR + 0x2364	Read/Write
Receive DMA & Scatter Gather Packet Wait Bound	C_DEV_BASEADDR + 0x2368	Read/Write
Receive DMA & Scatter Gather Interrupt Status Register	C_DEV_BASEADDR + 0x236C	Read/toggle on Write
Receive DMA & Scatter Gather Interrupt Enable Register	C_DEV_BASEADDR + 0x2370	Read/Write
Write Packet FIFO reset (write) Module Identification (read)	C_DEV_BASEADDR + 0x2000	Read/Write
Write Packet FIFO Vacancy	C_DEV_BASEADDR + 0x2004	Read
Write Packet FIFO 64 bit wide data write port	C_DEV_BASEADDR + 0x2100	Write
Read Packet FIFO reset (write) Module Identification (read)	C_DEV_BASEADDR + 0x2010	Read/Write
Read Packet FIFO Occupancy	C_DEV_BASEADDR + 0x2014	Read
Read Packet FIFO 64 bit wide data read port	C_DEV_BASEADDR + 0x2200	Read
Device Interrupt Status Register	C_DEV_BASEADDR + 0x0000	Read/Write
Device Interrupt Pending Register	C_DEV_BASEADDR + 0x0004	Read/Write
Device Interrupt Enable Register	C_DEV_BASEADDR + 0x0008	Read/Write
Device Interrupt Identification Register	C_DEV_BASEADDR + 0x0018	Read/Write
Device Global Interrupt Enable	C_DEV_BASEADDR + 0x001C	Read/Write
IP Interrupt Status Register	C_DEV_BASEADDR + 0x0020	Read/Write
IP Interrupt Enable Register	C_DEV_BASEADDR + 0x0028	Read/Write
Device Software Reset (write) Module Identification (read) Register	C_DEV_BASEADDR + 0x0040	Read/Write

GEMAC Core Registers

The GEMAC core registers are listed in Table 5. The SAH, SAL, CEAH, and CEAL are not currently used in this design. They are reserved for future use.

Table 5: **GEMAC Core Registers**

Register Name	PLB ADDRESS	Access
GEMAC Module Identification Register (GMIR)	C_DEV_BASEADDR + 0x1000	Read
GEMAC Control Register (GCR)	C_DEV_BASEADDR + 0x1004	Read/Write
Interframe Gap Register (IFGP)	C_DEV_BASEADDR + 0x1008	Read/Write
Station Address High (SAH)	C_DEV_BASEADDR + 0x100C	Read/Write
Station Address Low (SAL)	C_DEV_BASEADDR + 0x1010	Read/Write
MII Management Control Register (MGTCR)	C_DEV_BASEADDR + 0x1014	Read/Write
MII Management Data Register (MGTDR)	C_DEV_BASEADDR + 0x1018	Read/Write
Receive Packet Length Register (RPLR)	C_DEV_BASEADDR + 0x101C	Read
Transmit Packet Length Register (TPLR)	C_DEV_BASEADDR + 0x1020	Read/Write
Transmit Status Register (TSR)	C_DEV_BASEADDR + 0x1024	Read
Transmit Pause Packet Register (TPPR)	C_DEV_BASEADDR + 0x1028	Read/Write
CAM Entry Address High (CEAH)	C_DEV_BASEADDR + 0x102C	Read/Write
CAM Entry Address Low (CEAL)	C_DEV_BASEADDR + 0x1030	Read/Write
Receive Status Register (RSR)	C_DEV_BASEADDR + 0x1034	Read

GEMAC Module Identification Register (GMIR)

The GEMAC Version Register provides the software with a convenient method of verifying the GEMAC IP version and type.

.

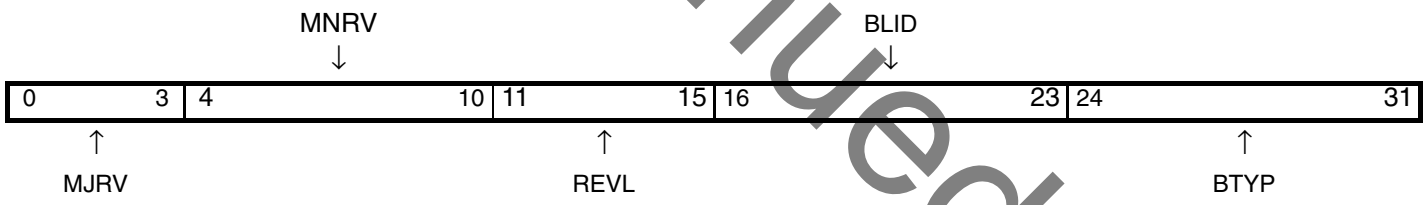

Figure 5: **GMIR**

Table 6: **GEMAC Module Identification Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 3	Major Version Number (MJRV)	Read	Version ID "0001" for this major version of 1	Module Major Version Number.
4 - 10	Minor Version Number (MNRV)	Read	Version ID "0000000" for this minor version of 0	**Module Minor Version Number.**
11 -15	Rev. Letter (REVL)	Read	Version ID "00000" for this revision of "a"	Module Minor Version Letter. This is a binary encoding of small case letters a through z (00000 - 11001).
16 - 23	Block ID (BLID)	Read	Assigned by Platform Generator defaults to "00000001"	**Block ID Number.** Distinct number for each GEMAC instantiated by Platform Generator.
24 - 31	Block Type (BTYP)	Read	TBD	**Block Type.** This is an 8 bit identifier unique to each IP type. For GEMAC this type is hex **TBD**.

GEMAC Control Register (GCR)

The GEMAC Control Register controls the operation of the GEMAC. Please note that some of these bits should not be changed while transmit and receive are enabled (GCR.ENTX and/or GCR.ENRX ='1').

.

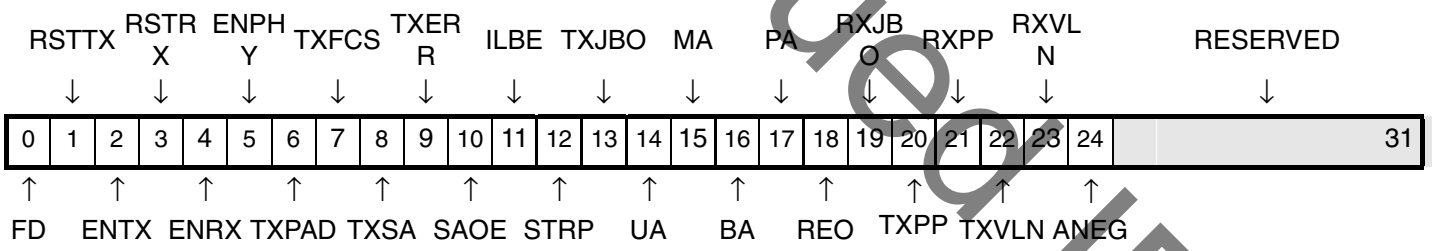

Figure 6: **GCR**

Table 7: **GEMAC Control Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0	FD	Read/Write	'0'	Full Duplex. Selects either full duplex mode (i.e., GEMAC can receive and transmit simultaneously on a dedicated Ethernet bus segment) or half duplex mode. This must be set to '1' (full duplex) for this design. Choosing half duplex enables CSMA/CD mode. Choosing full duplex mode disables CSMA/CD mode. It is the responsibility of the software to ensure that this mode matches the PHY if the PHY is operating in auto-negotiation mode. This bit should not be modified while transmit and receive are enabled GCR.ENTX and/or GCR.ENRX ='1'. '0' -Half Duplex '1' - Full Duplex
1	RSTTX	Read/Write	'1'	Reset Transmitter. Immediately resets the transmitter circuitry regardless of its current state. The transmitter circuitry will remain in reset until this bit is set to '0'. '0' - Normal Operation '1' - Reset
2	ENTX	Read/Write	'0'	Enable Transmitter. The transmitter circuitry will leave the idle state and begin transmission of a packet only when this bit is '1' and the transmit length register is not empty. Setting this bit to '0' will cause the transmitter to enter the idle state after completion of any packet transmission in progress (graceful halt). '0' - Disable Transmitter '1'- Enable Transmitter
3	RSTRX	Read/Write	'1'	Reset Receiver. Immediately resets the receiver circuitry regardless of its current state. The receiver circuitry will remain in reset until this bit is set to '0'. '0' - Normal Operation '1' - Reset
4	ENRX	Read/Write	'0'	Enable Receiver. The receiver circuitry will leave the idle state and begin monitoring the Ethernet bus only when this bit is '1'. Setting this bit to '0' will cause the receiver to enter the idle state after completion of any packet reception in progress (graceful halt). '0' - Disable Receiver '1'- Enable Receiver
5	ENPHY	Read/Write	'1'	Enable PHY. This value of this bit is driven to the PHY interface reset_n signal. If the external PHY supports this signal and this bit is '0', the PHY will reset and remain in reset until this bit is set to '1'. '0' - Disable / Reset PHY '1'- Enable PHY

Table 7: **GEMAC Control Register Bit Definitions *(Continued)***

Bit Location	Name	Core Access	Reset Value	Description
6	TXPAD	Read/Write	’1’	Enable Transmit Auto Pad Insertion. Enables automatic pad field insertion by the GEMAC circuitry if it is necessary. When this is enabled, the transmit packet data provided to the GEMAC should not contain pad data. When this is enabled, auto FCS insertion must also be selected to insure correct FCS calculation over the pad field. When this is disabled, the transmit packet data provided to the GEMAC should contain pad data if required. This bit should not be modified while transmit and receive are enabled GCR.ENTX =’1’ and/or GCR.ENRX =’1’. ’0’ - Disable Auto Pad Insertion ’1’- Enable Auto Pad Insertion
7	TXFCS	Read/Write	’1’	Enable Transmit Auto FCS Insertion. Enables automatic FCS field insertion by the GEMAC circuitry. When this is enabled, the transmit packet data provided to the GEMAC should not contain FCS data. When this is disabled, the transmit packet data provided to the GEMAC should contain FCS data. This bit should not be modified while transmit and receive are enabled GCR.ENTX =’1’ and/or GCR.ENRX =’1’. ’0’ - Disable Auto FCS Insertion ’1’- Enable Auto FCS Insertion
8	TXSA	Read/Write	’1’	Reserved
9	TXERR	Read/Write	’0’	Transmit Error Insertion. The value of this bit is driven to the PHY interface TX_ER signal. If the external PHY supports this mode, it will inject an error encoded byte into the transmit data. This bit should not be modified while transmit and receive are enabled GCR.ENTX =’1’ and/or GCR.ENRX =’1’. ’0’ - Disable Error Insertion ’1’ - Enable Error Insertion
10	SAOE	Read/Write	’1’	Reserved
11	ILBE	Read/Write	’0’	**Internal Loop-Back Enable.** Enables looping of the transmit data directly to the receive data path internally to the GEMAC. The transmit and receive paths are isolated from the external PHY.
12	STRP	Read/Write	’0’	Pad & FCS Strip Enable. Enables stripping of receive pad and FCS fields when type/length field is a length. ’0’ - Disable Strip ’1’ - Enable Strip
13	TXJBO	Read/Write	’0’	Jumbo Frame Transmit Enable. Enables transmission of jumbo frames. ’0’ - Disable Tx of jumbo frame ’1’ - Enable Tx of jumbo frame
14	UA	Read/Write	’1’	Reserved

Table 7: **GEMAC Control Register Bit Definitions *(Continued)***

Bit Location	Name	Core Access	Reset Value	Description
15	MA	Read/Write	'0'	Reserved
16	BA	Read/Write	'1'	Reserved
17	PA	Read/Write	'0'	Reserved
18	REO	Read/Write	'0'	Reserved
19	RXJBO	Read/Write	'0'	Jumbo Frame Receive Enable. Enables reception of jumbo frames. '0' - Disable Rx of jumbo frame '1' - Enable Rx of jumbo frame
20	TXPP	Read/Write	'0'	Transmit Pause Packets. Enables the GEMAC to transmit pause packets. '0' - Disable Tx Pause Packets '1' - Enable Tx Pause Packets
21	RXPP	Read/Write	'0'	Receive Pause Packets. Enables the GEMAC to process valid received pause packets. '0' - Disable Rx Pause Packets '1' - Enable Rx Pause Packets
22	TXVLN	Read/Write	'0'	Transmit VLAN Packets. Enables the GEMAC to transmit VLAN packets. '0' - Disable Tx VLAN Packets '1' - Enable Tx VLAN Packets
23	RXVLN	Read/Write	'0'	Receive VLAN Packets. Enables the GEMAC to receive VLAN packets. '0' - Disable Rx VLAN Packets '1' - Enable Rx VLAN Packets
24	ANEG	Read/Write	'1'	Enable Auto Negotiation. Enables the MGTs to perform auto negotiation. '0' - Disable auto negotiation '1' - Enable auto negotiation
23-31	Reserved	Read	0x00	**Reserved.** These bits are reserved for future use.

Interframe Gap Register (IFGP)

The Interframe Gap Register controls the duration of the interframe Gap. The Interframe Gap is measured in units of the bit time multiplied by eight. Please refer to the paragraph Interframe Gap and Deferring for information about how the Interframe Gap is used by the GEMAC. Please note that these settings should not be changed while transmit and receive are enabled (GCR.ENTX and/or GCR.ENRX ='1').

Figure 7: **IFGP**

Table 8: **Interframe Gap Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0-7	IFGP	Read/Write	"00001100"	Interframe Gap. A value of 1 in this field would provide an 8 bit time interframe gap. The reset value for this field is value recommended for most operation by IEEE Std. 802.3. This field should not be modified while transmit and receive are enabled GCR.ENTX ='1' and/or GCR.ENRX ='1'.
8-31	Reserved	Read	0x000000	**Reserved.** These bits are reserved for future use.

Receive Packet Length Register (RPLR)

The receive packet length register is actually a FIFO of register values each corresponding to a valid frame received. The data for the frame is stored in the receive data FIFO and the status word is stored in the receive status register FIFO.

The data is written by the GEMAC when the frame's destination address passes the current address validation modes and when the frame has been determined to be valid and the receive data FIFO had enough locations that all of the frame data has been saved.

The existence of data in the receive packet length FIFO (FIFO empty flag is '0') may be used to initiate the processing of received packets until this FIFO is empty. Reading this register causes the current value to be removed from the FIFO.

This register is wide enough to support 9K byte length jumbo frames (actually capable of up to 16K byte lengths).

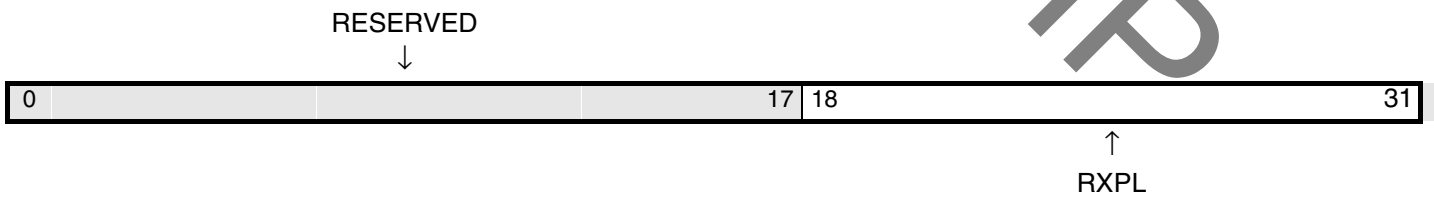

Figure 8: **RPLR**

Table 9: **Receive Packet Length Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0-17	Reserved	Read	0x00000	**Reserved.** These bits are reserved for future use.
18-31	RXPL	Read	0x0000	Receive Packet Length. The number of bytes of the corresponding receive packet stored in the receive data FIFO.

Transmit Packet Length Register (TPLR)

The transmit packet length register is actually a FIFO of register values each corresponding to a valid frame ready for transmit. The data for the frame is stored in the transmit data FIFO.

The data is written to the GEMAC over the external processor bus interface either by simple DMA, Scatter/Gather DMA, or by direct memory mapped access.

When presenting a transmit packet to the GEMAC, the packet data should first be written to the transmit data FIFO. The existence of data in the transmit packet length FIFO (FIFO empty flag is '0') is used by the GEMAC to initiate the processing of transmit packets until this FIFO is empty.

This register can be read over the processor interface but only the GEMAC can remove a value from the FIFO. The GEMAC will remove the current length from the FIFO when it completes the corresponding transmission. If multiple reads are performed prior to that completion, the same value will be returned for each read operation.

This register is wide enough to support 9K byte length jumbo frames (actually capable of up to 16K byte lengths).

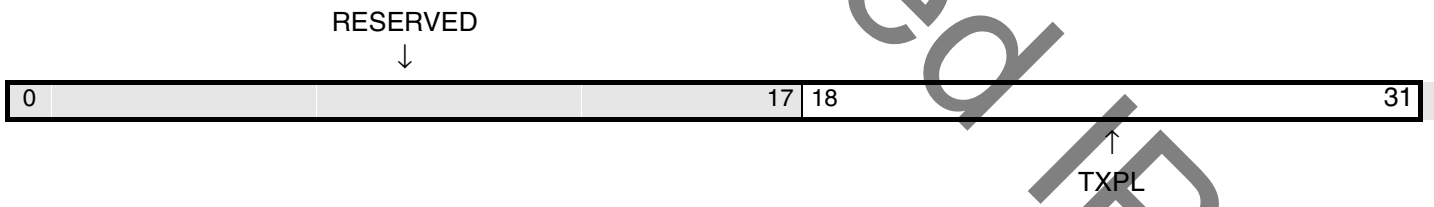

Figure 9: **TPLR**

Table 10: **Transmit Packet Length Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0-17	Reserved	Read	0x00000	**Reserved.** These bits are reserved for future use.
18-31	TXPL	Read/Write	0x0000	Transmit Packet Length. The number of bytes of the corresponding transmit packet stored in the transmit data FIFO.

Receive Status Register (RSR)

The receive status register is a place holder for the receive status register that is used by the Scatter Gather DMA interface. The GEMAC does not need a receive status register but is required to provide the correct value in bit 31 to the generalized Scatter Gather DMA circuitry as part of a standard receive packet operation.

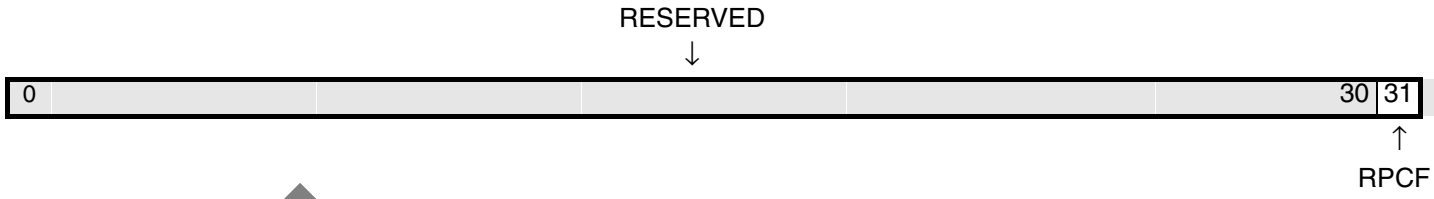

Figure 10: **RSR**

Table 11: **Receive Status Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 30	Reserved	N/A	0x00000000	Reserved. These bits are unused and will always return all zeros.
31	RPCF	Read	'1'	Receive Packet Complete Flag. This bit is always '1' and is used to indicate to the software that a buffer descriptor associated with this packet has been completely processed by the DMA circuitry and is available for software use.

Transmit Status Register (TSR)

The transmit status register is actually a FIFO of register values each corresponding to a frame transmission attempt. The bits in this register reflect the specific status of the corresponding transmit operation including the GEMAC settings which were applied to the transmit operation. Reading this register causes the current value to be removed from the FIFO.

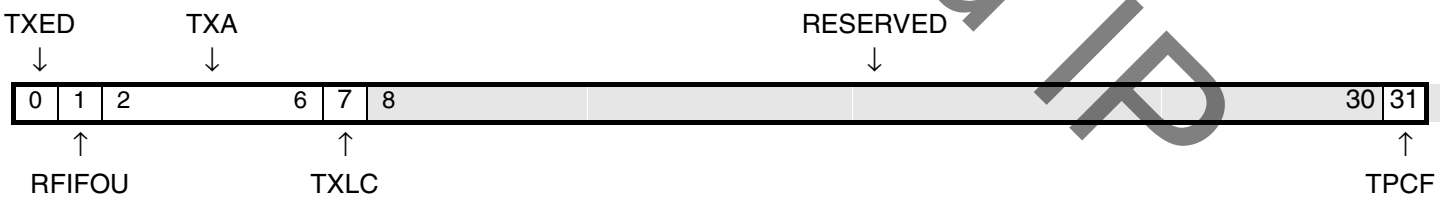

Figure 11: **TSR**

Table 12: **Transmit Status Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0	TXED	Read	'0'	Transmit Excess Deferral Error. This bit is only applicable in half-duplex mode. It indicates that at least one transmit frame was not able to complete transmission due to collisions that exceed the maximum number of retries (16). This bit is cleared to '0' when read. '0' - No excess deferrals occurred since the last read '1' - At least one excess deferral has occurred
1	PFIFOU	Read	'0'	Packet Fifo Underrun. This bit indicates that at least one transmit frame experienced a packet FIFO underrun condition during transmission. This bit is cleared to '0' when read. '0' - No packet FIFO underruns occurred since the last read '1' - At least one packet FIFO underrun has occurred
2- 6	TXA	Read	0x00	**Transmission Attempts.** The number of transmission attempts made. There will be a maximum of 16 attempts.
7	TXLC	Read	'0'	Transmit Late Collision Error. This bit is only applicable in half-duplex mode. It indicates a non-recoverable collision occurred more than 512-byte times after the start of the transmission. No automatic retransmission can be attempted by the GEMAC. A late collision should never occur on a compliant Ethernet network. '0' - No late collisions occurred '1' - Late collision occurred
8 - 30	Reserved	N/A	0x000000	Reserved. These bits are unused and will always return all zeros.
31	TPCF	Read	'1'	Transmit Packet Complete Flag. This bit is always '1' and is used to indicate to the software that a buffer descriptor associated with this packet has been completely processed by the DMA circuitry and is available for software use.

Station Address High Register (SAH)

This register contains the high-order 16 bits of the 48 bit station address.

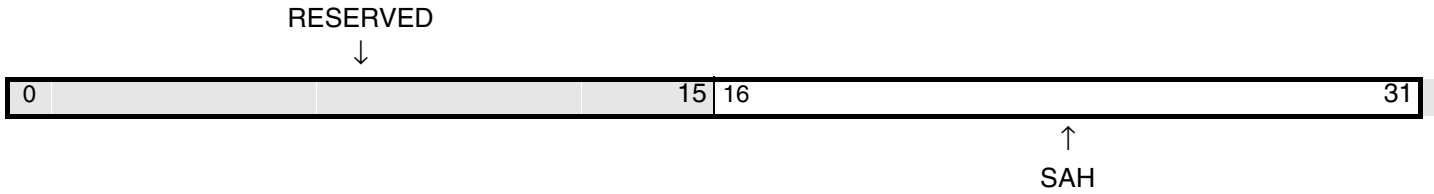

Figure 12: **SAH**

Table 13: **Station Address High Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0-15	Reserved	Read	0x0000	**Reserved.** These bits are reserved for future use.
16-31	SAH	Read/Write	0x0000	Station Address High. This register should not be modified while transmit and receive are enabled GCR.ENTX =’1’ and/or GCR.ENRX =’1’

Station Address Low Register (SAL)

This register contains the low-order 32 bits of the 48 bit station address.

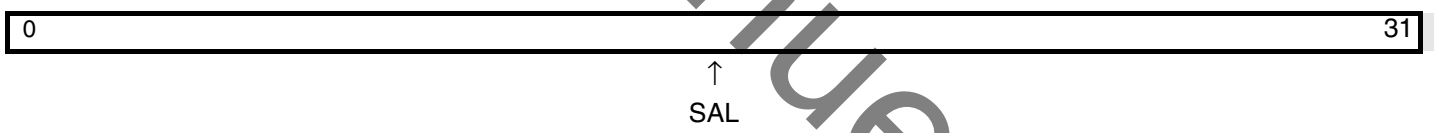

Figure 13: **SAL**

Table 14: **Station Address Low Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0-31	D0 - D31	Read/Write	0x00000000	Station Address Low. This register should not be modified while transmit and receive are enabled GCR.ENTX =’1’ and/or GCR.ENRX =’1”

MII Management Control Register (MGTCR)

The MII management control register is used with the MII management data register to perform read and writes between the GEMAC and the PHY device via the MII management interface.

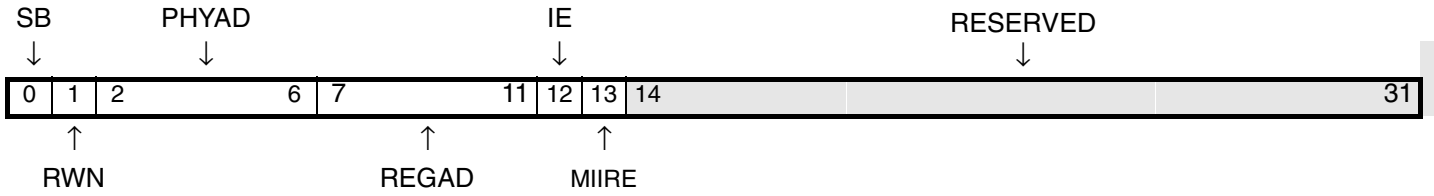

Figure 14: **MGTCR**

Table 15: **MII Management Control Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0	SB	Read/Write	’0’	Start / Busy. writing a ’1’ to this bit initiates an MII read or write operation. The GEMAC will clear this bit to ’0’ when the operation has been completed. ’0’ - No MII Operation in Progress ’1’- MII Read or Write in Progress
1	RWN	Read/Write	’1’	Read Write Not. This bit indicates the direction of the MII operation. ’0’ - Write to PHY register ’1’- Read from PHY register
2-6	PHYAD	Read/Write	0x00	PHY Address. This field is used to specify the address of the PHY to be accessed.
7-11	REGAD	Read/Write	0x00	**Register Address.** This field is used to specify the register in the PHY to be accessed.
12	IE	Read/Write	’0’	MII Management Interface Enable. This bit controls the 3-state drivers for the MII management signal interface to the PHY. ’0’ - The MII management signals to the PHY are 3-stated. ’1’ - The MII management signals to the PHY are driven and controlled by the GEMAC management interface.
13	MIIRE	Read	’0’	MII Management Read Error. Indicates that a read from a PHY register is invalid and the operation should be retried. This is indicated during a read turn-around cycle when the PHY does not drive the MDIO signal to the low state. This bit is cleared to ’0’ when read. ’0’ - No read errors occurred since the last read ’1’ - At least one read error has occurred
14-31	Reserved	Read	0x00000	**Reserved.** These bits are reserved for future use.

MII Management Data Register (MGTDR)

The MII management data register is used with the MII management control register to perform read and writes between the GEMAC and the PHY device via the MII management interface. For a PHY register write operation, data should be written to the data register prior to the write to the control register.

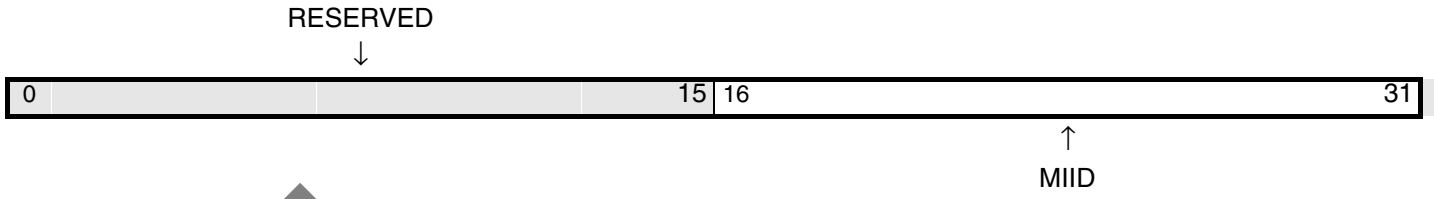

Figure 15: **MGTDR**

Table 16: **MII Management Data Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0 - 15	Reserved	N/A	0x0000	Reserved. These bits are unused and will always return all zeros.
16-31	MIID	Read/Write	0x0000	MII Management Data Register.

Transmit Pause Packet Register (TPPR)

This register value is used to transmit a automatically formed pause packet when requested via a bit in the control register. This value represents the number of pause quanta (512 bit times) during which the receiving MAC should pause transmissions. A write to this register initiates the tranmission of a pause packet if enabled in the control register (GCR).

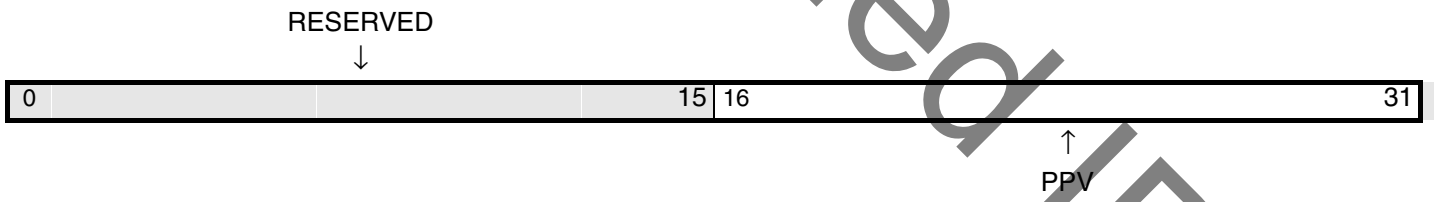

Figure 16: **TPPR**

Table 17: **Transmit Pause Packet Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0-15	Reserved	Read	0x0000	**Reserved.** These bits are reserved for future use.
16-31	PPV	Read/Write	0x0000	Pause Packet Value. This register should not be modified while transmit and receive are enabled GCR.ENTX =’1’ and/or GCR.ENRX =’1’

CAM Entry Address High Register (CEAH)

This register is reserved for future use.

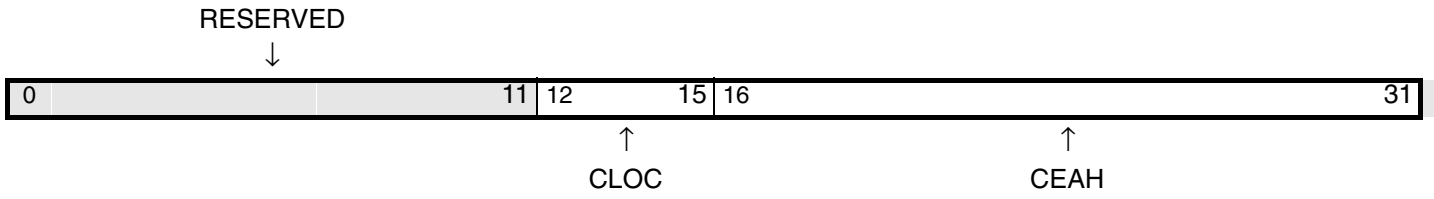

Figure 17: **CEAH**

Table 18: **CAM Entry Address High Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0-11	Reserved	Read	0x0000	**Reserved.** These bits are reserved for future use.
12-15	CLOC	Read/Write	0000	**CAM Entry Location.** These bits are used to indicate which of the 16 CAM locations is to be updated.
16-31	CEAH	Read/Write	0x0000	CAM Entry Address High. This register should not be modified while transmit and receive are enabled GCR.ENTX =’1’ and/or GCR.ENRX =’1’

CAM Entry Address Low Register (CEAL)

This register is reserved for future use.

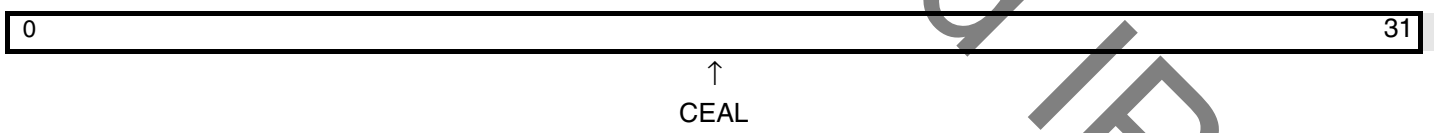

Figure 18: **CEAL**

Table 19: **CAM Entry Address Low Register Bit Definitions**

Bit Location	Name	Core Access	Reset Value	Description
0-31	D0 - D31	Read/Write	0x00000000	CAM Entry Address Low. This register should not be modified while transmit and receive are enabled GCR.ENTX =’1’ and/or GCR.ENRX =’1”

MII Management Interface

MII Management Clock

The Management Data Clock (MDC) is driven by the GEMAC to the PHY as the reference for data transfer on the MDIO signal. This signal has no maximum high and low times and need not be periodic but must have a minimum high and low time of at least 160 nS with a corresponding minimum period of 400 nS (corresponds to a maximum of 2.5 MHz) in order to comply with the IEEE 802.3-2000 specification for this interface.

The MII clock frequency is derived from the PLB clock using the C_MIIM_CLKDVD generic of and is defined as PLB_Clk / [(CLKDVD + 1) * 2]. For a 100 Mhz PLB clock, a C_MIIM_CLKDVD value of "10011" generates a 2.5 Mhz MDC.

The maximum C_MIIM_CLKDVD value of "11111" generates a 2.5 Mhz MDC from a PLB clock of 160 Mhz.

MII Management Data

The Management Data Input/Output signal (MDIO) is a bidirectional signal between the PHY and the GEMAC used to transfer control and status. All transfers via the MDIO are synchronous to the MDC signal. MDIO must driven through 3-state pins that enable either the GEMAC or the PHY to drive the signal.

The necessary pull-up and pull-down resistors for this signal are defined in the IEEE Std. 802.3 paragraph 22.4.4.2. When the GEMAC drives this signal, it must provide a minimum of 10nS setup and hold in reference to the MDC signal. When the PHY drives this signal, it will have a clock to output delay of 0 to 300nS.

Management Frame Structure

Data transferred via the MDIO is structured as frames as shown in Table 20.

Table 20: **Management Frame Format**

	Management Frame Fields							
	PRE	**ST**	**OP**	**PHYAD**	**REGAD**	**TA**	**DATA**	**IDLE**
Read	1...1	01	10	AAAAA	RRRRR	Z0	DDDDDDDDDDDDDDDD	Z
Write	1...1	01	01	AAAAA	RRRRR	10	DDDDDDDDDDDDDDDD	Z

Idle

The IDLE condition on MDIO is a high-impedance state. All 3-state drivers are disabled and the PHY's pull-up resistor will pull the MDIO line to a logic one.

Preamble (PRE)

At the beginning of each management transaction, the GEMAC will transmit 32 contiguous logic one bits on the MDIO signal.

Start of Frame (ST)

The GEMAC follows the preamble with the start of frame pattern which is "01".

Operation Code (OP)

The operation code is by the GEMAC following the start of frame to indicate a read or write transaction. A read is designated by "10" and a write by "01".

PHY Address (PHYAD)

The next field transmitted by the GEMAC is the 5 bit PHY address field. This identifies one of up to 32 PHYs connection to the same interface. The most significant bit of the address is transmitted first. The special PHY address "00000" will address any PHY.

Register Address (REGAD)

The GEMAC follows the PHY address field with the transmission of the 5 bit register address field allowing the access of up to 32 PHY registers. The most significant bit of the address is transmitted first.

Turn Around (TA)

A turn around period of 2 bit times follows the register field and precedes the data field to prevent contention during a read cycle. For a read, both the GEMAC and the PHY remain in a high-impedance state for the first bit time of the turnaround. The PHY will drive a zero bit during the second bit time of the turnaround of a read transaction.

During a write, the GEMAC must drive a one bit for the first bit time of the turnaround and a zero bit for the second bit time of the turnaround. Figure 19 shows the behavior of the MDIO signal during the turnaround field of a read transaction.

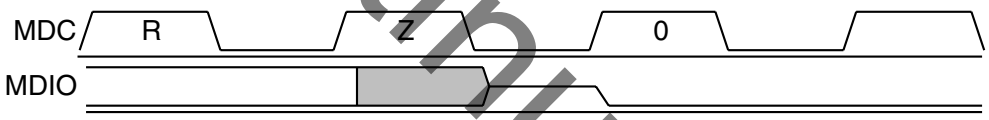

Figure 19: **MDIO Read Turn Around**

Data

The data field is 16 bits and is transmitted and received with bit 15 (MSb) first of the register being addressed.

Optional GMII

The GMII is designed to IEEE 802.3-2000 Clause 35. This utilizes IOB's of the Xilinx devices, using input/output buffers and input/output flip-flops to maximize setup and hold margins. All IOB logic is placed in the top level wrapper. All clock management logic for the core is placed in this wrapper; this allows multiple instances of the core to share clocks.

Clock Management

Figure 20 illustrates the clock management used with the GMII interface. All clocks shown have a frequency of 125MHz.

GTX_CLK must be provided to the GEMAC core. This is a high quality clock which satisfies the IEEE802.3-2000 requirements. It is expected that this clock will be derived from an external oscillator and connected into the device through an IBUFG as illustrated. The clock source must then be routed onto a global clock network by connecting it to a BUFG.

Within the GEMAC core, this global clock is used by all MAC transmitter logic. GMII_TX_CLK is derived from this global clock by inverting it and routing it to an OBUF.

GMII_RX_CLK is received through an IBUFG. This clock is then routed onto a global clock network by connecting it to a BUFG. The resulting global clock is used by all MAC receiver logic.

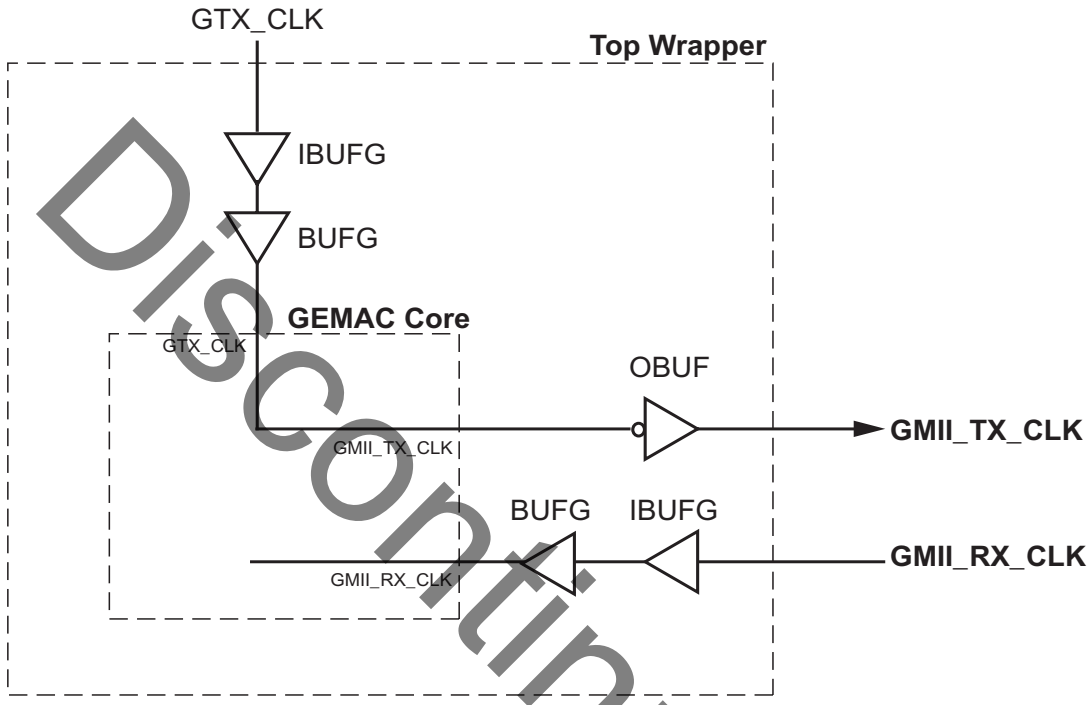

Figure 20: **Clock Management with GMII**

Optional PCS Sublayer with Ten-Bit Interface (TBI) for 1000BASE-X

The Full-Duplex Physical Coding Sublayer (PCS) for 1000BASE-X is defined in IEEE 802.3-2000 Clauses 36 and 37.

A block diagram of the PCS sublayer can be seen in Figure 21 The major functional blocks of the PCS sublayer are:

- the Transmit Engine,
- the Auto-Negotiation Block,
- the Receive Engine and Synchronization Block,
- the PCS Managed Register Block,
- the 8B/10B Encoder Block,
- the 8B/10B Decoder Block, and
- the Receiver Elastic Buffer.

The PCS sublayer's Managed Register Block is accessed through the MDIO Interface as if it were an externally connected PHY. This configures the operation of the PCS sublayer and Auto-Negotiation.

The 8B/10B encoding/decoding blocks are implemented by configuring Block RAMs as ROM. These are used as large look-up tables. The remaining PCS sublayer functionality such as code group encoding/decoding, the Managed Register Block, and Auto-Negotiation are implemented in CLB logic.

The Receiver Elastic buffer enables the 10-bit parallel data received from the PMA sublayer on the PMA_RX_CLK0 / PMA_RX_CLK1 to be transferred onto the internal 125MHz clock domain derived from GTX_CLK. This buffer is implemented in Distributed Memory.

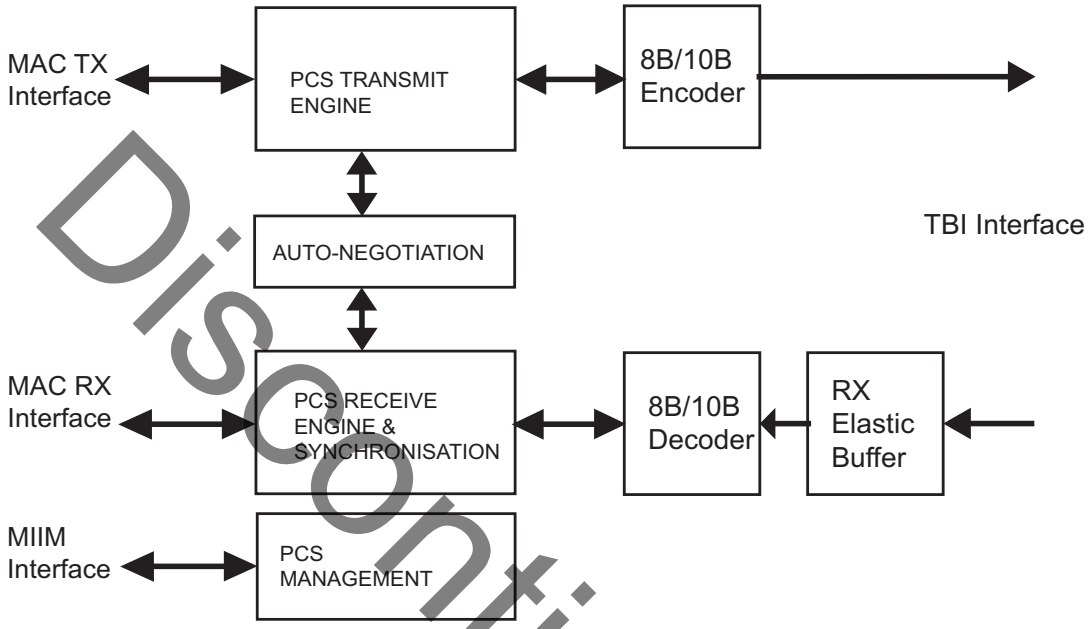

Figure 21: **Functional Block Diagram of PCS sublayer with TBI interface**

TBI Interface

The TBI is designed to IEEE 802.3-2000 Clause 36.3. This utilizes IOB's of the Xilinx devices, using input/output buffers and input/output flip-flops to maximize setup and hold margins. All IOB logic is placed in the top level wrapper. In addition, all clock management logic for the core is placed in this wrapper; this allows multiple instances of the core to share clocks.

Clock Management

Figure 22 illustrates the clock management used with the TBI interface.

GTX_CLK must be provided to the MAC core. This is a high quality clock of frequency 125MHz which satisfies the IEEE802.3-2000 requirements. It is expected that this clock will be derived from an external oscillator and connected into the device through an IBUFG as illustrated. The clock source must then be routed onto a global clock network by connecting it to a BUFG.

Within the GMAC core, this global clock is used by all MAC transmitter logic. PMA_TX_CLK is derived from this global clock by inverting it and routing it to an OBUF.

PMA_RX_CLK0 and PMA_RX_CLK1 are each received through an IBUFG. These clocks are then routed onto global clock networks by connecting each to a BUFG. PMA_RX_CLK0 and PMA_RX_CLK1 are each 62.5MHz clocks.

Data from the TBI is received alternatively on the rising edge of PMA_RX_CLK0 followed by the rising edge of PMA_RX_CLK1 using IOB Double-Data Rate input registers. This data is then written into the Receiver Elastic Buffer. Data is read out of the Elastic Buffer using the global clock derived from GTX_CLK; all the remaining receiver logic is synchronous to this clock.

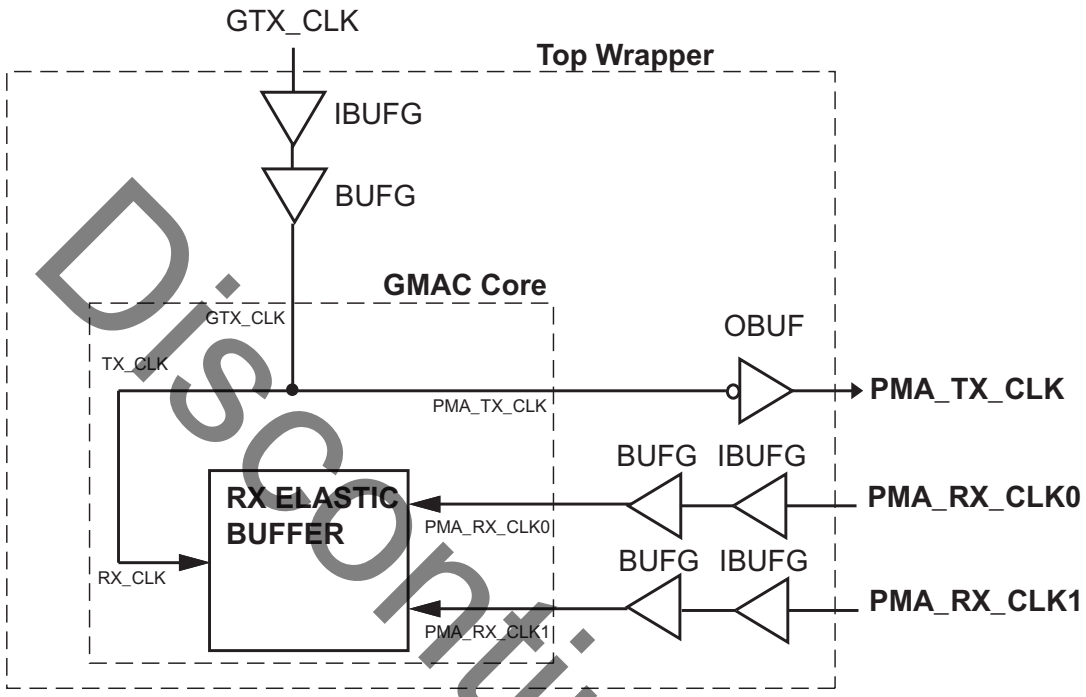

Figure 22: **Clock Management for PCS sublayers with TBI Interface**

Optional PCS and PMA Sublayers for 1000BASE-X

The optional Full-Duplex Physical Coding Sublayer (PCS) with Physical Medium Attachment (PMA) for 1000BASE-X is defined in IEEE 802.3-2000 Clauses 36 and 37. This functionality is only available in the Virtex™II Pro family of FPGA devices.

A block diagram of the PCS and PMA sublayers can be seen in Table 23 The functional blocks of the PCS and PMA sublayers may replace the optional GMII interface. The functional blocks of the PCS and PMA sublayers are:

- The Transmit Engine
- The Auto-Negotiation Block
- The Receive Engine and Synchronization Block
- The PCS Managed Register Block
- The Rocket I/O Multi-Gigabit Transceiver

The PCS sublayer's Managed Register Block is accessed through the MDIO Interface as if it were an externally connected PHY. This configures the operation of the PCS sublayer, PMA sublayer, and Auto-Negotiation.

The Rocket I/O Multi-Gigabit Transceiver (MGT) provides some of the PCS layer functionality such as 8B/10B encoding/decoding and the PMA serdes. The remaining PCS sublayer functionality such as code group encoding/decoding, the Managed Register Block, and Auto-Negotiation are implemented in CLB logic.

The Rocket I/O Multi-Gigabit Transceiver provides a programmable serial output differential swing of 5 levels ranging between 800 mV and 1600 mV; on-chip termination impedance of 50 ohms or 70 ohms; 4 levels of programmable pre-emphasis. These can be set from the User Constraints File (UCF).

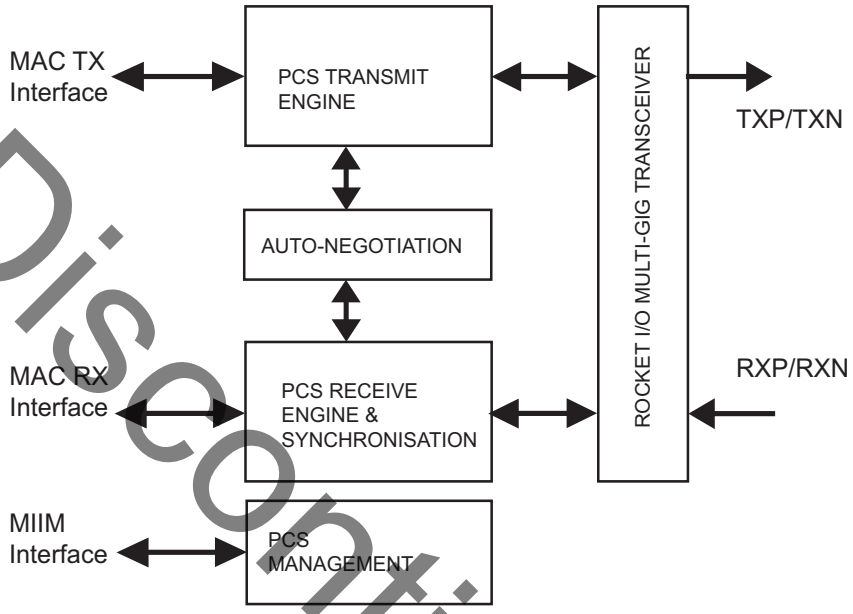

Figure 23: **Functional Block Diagram of PCS/PMA Sublayer**

Clock Management

Table 24 illustrates the clock management that should be used with the optional PCS and PMA sublayers for 1000BASE-X. This logic is implemented in the top level wrapper. This provides flexibility since the design can be modified to use either REFCLK or BREFCLK (please refer to Rocket I/O documentation). It would also allows multiple instances of the core to share clocks.

(B)REFCLK must be provided to the MAC core. This clock must be derived from an external oscillator and connected to the core through an IBUFG as illustrated. (B)REFCLK is a high quality clock of frequency 62.5 MHz which satisfies the Rocket I/O Multi-Gigabit Transceiver (MGT) requirements.

(B)REFCLK is routed to both the core and a DCM. From the DCM the CLK2X180 output is routed to a BUFG to provide the device with a global 125 MHz clock (connected to USERCLK2 input on the core). Within the core, both TX_CLK and RX_CLK share this global 125 MHz clock source, which is used by all transmitter and receiver logic. This is made possible by the RX_ELASTIC_BUFFER functionality of the Rocket I/O MGT (refer to Virtex™II Pro documentation). The MAC Client can obtain this clock by connecting to either the TX_CLK or the RX_CLK signals output from the core.

From the DCM the CLK0 output is routed to a BUFG to provide the device with a global 62.5MHz clock. This is connected to the USRCLK input of the core.

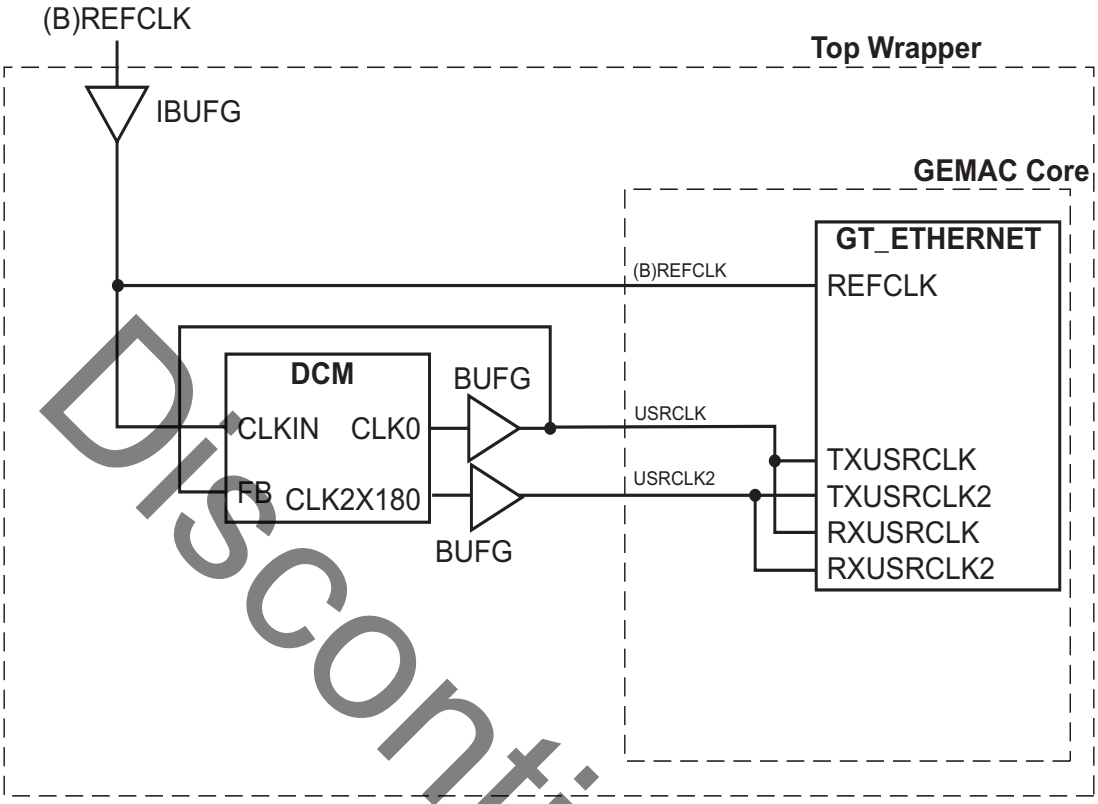

Figure 24: **Clock Management with PCS/PMA sublayers**

PCS Sublayer Managed Register Block

The PCS sublayer in both the TBI interface and the PCS/PMA (using the Virtex™II Pro Rocket I/O) versions contain the full Managed Register Block defined in IEEE 802.3-2000 clause 37. This utilizes ten dedicated management registers, accessed via an MDIO interface which is internally connected to that of the MAC. These are defined in Table 21 to Table inclusive.

.

Table 21: **Control Register (Register 0)**

Bit(s)	Name	Description	Attributes	Default Value
0.15	Reset	1 = PCS/PMA reset 0 = Normal Operation	Read/Write Self Clearing	0
0.14	Loopback	1 = Enable Loopback Mode 0 = Disable Loopback Mode	Read/Write	0
0.13	Speed Selection (LSB)	The core shall always return a 0 for this bit. Together with bit 0.6, speed selection of 1000 Mb/s is identified.	Returns 0	0
0.12	Auto-Negotiation Enable	1 = Enable Auto-Negotiation Process 0 = Disable Auto-Negotiation Process	Read/Write	1

Table 21: **Control Register (Register 0) *(Continued)***

Bit(s)	Name	Description	Attributes	Default Value
0.11	Power Down	1 = Power down 0 = Normal operation When set to '1' the Rocket I/O MGT is placed in a low power state. This bit requires a reset (see bit 0.15) to clear.	Read/ Write	0
0.10	Isolate	1 = Electrically Isolate PHY from GMII 0 = Normal operation	Read/Write	1
0.9	Restart Auto-Negotiation	1 = Restart Auto-Negotiation Process 0 = Normal Operation	Read/Write Self Clearing	0
0.8	Duplex Mode	The core shall always return a 1 for this bit to signal Full-Duplex Mode.	Returns 1	1
0.7	Collision Test	The core shall always return a 0 for this bit to disable COL test.	Returns 0	0
0.6	Speed Selection(MSB)	The core shall always return a 1 for this bit. Together with bit 0.13, speed selection of 1000 Mb/s is identified.	Returns 1	1
0.5:0	Reserved	Always returns 0's, writes ignored	Returns 0's	000000

Table 22: **Status Register (Register 1)**

Bit(s)	Name	Description	Attributes	Default Value
1.15	100BASE-T4	The core shall always return a 0 for this bit since 100BASE-T4 is not supported.	Returns 0	0
1.14	100BASE-X Full Duplex	The core shall always return a 0 for this bit since 100BASE-X Full Duplex is not supported.	Returns 0	0
1.13	100BASE-X Half Duplex	The core shall always return a 0 for this bit since 100BASE-X Half Duplex is not supported.	Returns 0	0
1.12	10 Mb/s Full Duplex	The core shall always return a 0 for this bit since 10 Mb/s Full Duplex is not supported.	Returns 0	0
1.11	10 Mb/s Half Duplex	The core shall always return a 0 for this bit since 10 Mb/s Half Duplex is not supported.	Returns 0	0
1.10	100BASE-T2 Full Duplex	The core shall always return a 0 for this bit since 100BASE-T2 Full Duplex is not supported.	Returns 0	0
1.9	100BASE-T2 Half Duplex	The core shall always return a 0 for this bit since 100BASE-T2 Half Duplex is not supported.	Returns 0	0
1.8	Extended Status	The core shall always return a 1 for this bit to indicate the presence of the Extended Register (Register 15).	Returns 1	1

Table 22: **Status Register (Register 1) *(Continued)***

Bit(s)	Name	Description	Attributes	Default Value
1.7	Reserved	Always return 0, writes ignored	Returns 0	0
1.6	MF Preamble Suppression	The core shall always return a 1 for this bit to indicate that Management Frame Preamble Suppression is supported.	Returns 1	1
1.5	Auto-Negotiation Complete	1 = Auto-Negotiation process completed 0 = Auto-Negotiation process not completed	Read Only	0
1.4	Remote Fault	1 = Remote fault condition detected 0 = No remote fault condition detected	Read Only Self Clearing on read	0
1.3	Auto-Negotiation Ability	The core shall always return a 1 for this bit to indicate that the PHY is capable of Auto-Negotiation.	Returns 1	1
1.2	Link Status	1 = Link is up 0 = Link is down	Read Only Self Clearing on read	0
1.1	Jabber Detect	The core shall always return a 0 for this bit since Jabber Detect is not supported.	Returns 0	0
1.0	Extended Capability	The core shall always return a 0 for this bit since no extended register set is supported.	Returns 0	0

Table 23: **PHY Identidyer (Registers 2 and 3)**

Bit(s)	Name	Description	Attributes	Default Value
2.15:0	Organizationally Unique Identifier	Always returns 0's	Returns 0's	0000000000000000
3.15:10	Organizationally Unique Identifier	Always returns 0's	Returns 0's	000000
3.9:4	Manufacturer's model number	Always returns 0's	Returns 0's	000000
3.3:0	Revision Number	Always returns 0's	Returns 0's	0000

Table 24: **Auto-Negotiation Advertisement Register (Register 4)**

Bit(s)	Name	Description	Attributes	Default Value
4.15	Next Page	1 = Next Page functionality is advertised 0 = Next Page functionality is not advertised	Read/Write	0
4.14	Reserved	Always returns 0, writes ignored	Returns 0	0
4.13:12	Remote Fault	00 = No Error 01 = Offline 10 = Link Failure 11 = Auto-Negotiation Error	Read/Write Self clearing to 00 after Auto-Negotiation	00
4.11:9	Reserved	Always return 0, writes ignored	Returns 0	0
4.8:7	Pause	00 = No PAUSE 01 = Asymmetric PAUSE towards link partner 10 = Symmetic PAUSE 11 = Both Symmetric PAUSE and Asymmetric PAUSE towards link partner	Read/Write	11
4.6	Half Duplex	The core shall always return a 0 for this bit since Half Duplex Mode is not supported.	Returns 0	0
4.5	Full Duplex	1 = Full Duplex Mode is advertised 0 = Full Duplex Mode is not advertised	Read/Write	1
4.4:0	Reserved	Always returns 0's, writes ignored	Returns 0's	00000

Table 25: **Auto-Negotiation Link Partner Ability Base Register (Register 5)**

Bit(s)	Name	Description	Attributes	Default Value
5.15	Next Page	1 = Next Page functionality is supported 0 = Next Page functionality is not supported	Read Only	0
5.14	Acknowledge	Used by Auto-Negotiation function to indicate reception of a link partner's base or next page.	Read Only	0
5.13:12	Remote Fault	00 = No Error 01 = Offline 10 = Link Failure 11 = Auto-Negotiation Error	Read Only	00
5.11:9	Reserved	Always returns 0's	Returns 0's	000
5.8:7	Pause	00 = No PAUSE 01 = Asymmetric PAUSE supported 10 = Symmetic PAUSE supported 11 = Both Symmetric PAUSE and Asymmetric PAUSE supported	Read Only	00

Table 25: **Auto-Negotiation Link Partner Ability Base Register (Register 5) *(Continued)***

Bit(s)	Name	Description	Attributes	Default Value
5.6	Half Duplex	1 = Half Duplex Mode is supported 0 = Half Duplex Mode is not supported	Read Only	0
5.5	Full Duplex	1 = Full Duplex Mode is supported 0 = Full Duplex Mode is not supported	Read Only	0
5.4:0	Reserved	Always returns 0's	Returns 0's	00000

Table 26: **Auto-Negotiation Expansion Register (Register 6)**

Bit(s)	Name	Description	Attributes	Default Value
6.15:3	Reserved	Always returns 0's	Returns 0's	0000000000000
6.2	Next Page Able	The core shall always return a 1 for this bit since the device is Next Page Able.	Returns 1	1
6.1	Page Received	1 = A new page has been received 0 = A new page has not been received	Read Only Self Clearing on read	0
6.0	Reserved	Always returns 0's	Returns 0's	0000000

Table 27: **Auto-Negotiation Next Page Transmit Register (Register 7)**

Bit(s)	Name	Description	Attributes	Default Value
7.15	Next Page	1 = Additional Next Page(s) will follow 0 = Last page	Read/Write	0
7.14	Reserved	Always returns 0	Returns 0	0
7.13	Message Page	1 = Message Page 0 = Unformatted Page	Read/Write	1
7.12	Acknowledge 2	1 = Will comply with message 0 = Cannot comply with message	Read/Write	0
7.11	Toggle	Value will toggle between subsequent Next Pages.	Read Only	0
7.10:0	Message / Unformatted Code Field	Message Code Field or Unformatted Page Encoding as dictated by 7.13.	Read/Write	00000000001 (Null Message Code)

Table 28: **Auto-Negotiation Next Page Receive Register (Register 8)**

Bit(s)	Name	Description	Attributes	Default Value
8.15	Next Page	1 = Additional Next Page(s) will follow 0 = Last page	Read Only	0
8.14	Acknowledge	Used by Auto-Negotiation function to indicate reception of a link partner's base or next page.	Read Only	0
8.13	Message Page	1 = Message Page 0 = Unformatted Page	Read Only	0
8.12	Acknowledge 2	1 = Will comply with message 0 = Cannot comply with message	Read Only	0
8.11	Toggle	Value will toggle between subsequent Next Pages.	Read Only	0
8.10:0	Message / Unformatted Code Field	Message Code Field or Unformatted Page Encoding as dictated by 8.13.	Read Only	00000000000

Table 29: **Extended Status Register (Register 15)**

Bit(s)	Name	Description	Attributes	Default Value
15.15	1000BASE-X Full Duplex	The core shall always return a 1 for this bit since 1000BASE-X Full Duplex is supported.	Returns 1	1
15.14	1000BASE-X Half Duplex	The core shall always return a 0 for this bit since 1000BASE-X Half Duplex is not supported.	Returns 0	0
15.13	1000BASE-T Full Duplex	The core shall always return a 0 for this bit since 1000BASE-T Full Duplex is not supported.	Returns 0	0
15.12	1000BASE-T Half Duplex	The core shall always return a 0 for this bit since 1000BASE-T Half Duplex is not supported.	Returns 0	0
15:11:0	Reserved	Always returns 0's	Returns 0's	000000000000

Packet FIFO Interface

The GEMAC uses the special "Mark", "Release", and "Restore" features of the packet FIFOs to assist in management of transmit and receive data under special circumstances.

The data ports of these FIFOs are 64 bits wide.

The use of the special features is discussed here as it applies to Ethernet operation. Also discussed are special considerations required to support a multi-cycle "dead" FIFO access time following a "Mark", "Release", and "Restore" operation.

Transmit Packet FIFO Interface

The use of the special FIFO features greatly simplifies the retransmission of a packet following a collision on a half duplex bus (the features are not needed for full duplex operation). When the GEMAC is ready to transmit (transmit enable is '1' and the transmit length register is not empty) and it is in half-duplex mode and it is not deferring, it will perform a "Mark" in the transmit FIFO and begin reading transmit data (after a several cycle delay required by the FIFO) while counting down the length value.

If no collision occurs after 512 byte times, the GEMAC will perform a "Release" to allow the FIFO locations containing the first parts of the transmit to be reused. Since the GEMAC will not be able to read during the several cycles following the "release", it will have to insure that the transmit bus FIFO contains enough nibbles to sustain continuous transmission flow until packet FIFO reads can be resumed.

If a collision does occur in the first 512 byte times, the GEMAC will transmit the JAM pattern, increment the retry count, and if retrys are less than 16, perform backoff and a "Restore" on the packet FIFO. Since the GEMAC will be performing backoff and deferral it will not need to access the FIFO during the "Dead" FIFO access time following the "Restore".

If a transmission was attempted 16 retries and was unsuccessful, the GEMAC will perform a "Release" and flush the remainder of the failed packet data from the transmit packet FIFO by performing the number of reads required to decrement the length counter to zero.

Receive Packet FIFO Interface

The use of the special FIFO features greatly simplifies the flushing of a packet following a receive error. When the GEMAC detects that a packet is being received, the GEMAC will perform a "Mark" on the receive packet FIFO and begin storing receive data (after a several cycle delay required by the FIFO) while counting up the length value.

The GEMAC will continue reception and will store data into the receive FIFO until either the reception is successful, a receive error occurs, or the receive packet FIFO becomes full and overruns.

Following the completion of a successful receive, the GEMAC will perform a "Release". However, if an error occurs during reception or if the receive packet FIFO becomes full and overruns, the GEMAC will discontinue data storage into the receive packet FIFO and will flush all packet data stored there associated with the bad packet by performing a "Restore". There should be not be a need for special handling of the "dead" FIFO access time for "Restore" operations.

Receive Address Validation

Destination addresses are classified as either unicast (a single station address indicated by the I/G bit ='0'), multicast (a group of stations indicated by the I/G bit ='1'), and the multicast subgroup broadcast (all stations on the network).

This GEMAC does not check destination addresses. It accepts all addresses.

Freeze Mode

The freeze mode input signal is used for debug purposes and represents a request for a graceful halt. When this signal is activated, the transmit, receive, and MII management interface circuitry will complete any operations in progress and will go to an idle state. without modifying the state of any internal registers except for those required at the end of the completed operations.

While the GEMAC is frozen, all registers normal accessible by the processor will remain accessible. When the freeze mode signal is deactivated, the GEMAC will resume operation in the mode it was in prior to freezing. Note this may cause the receive circuitry to resume operation while a receive frame is already in progress creating possible error conditions.

Error Handling

Transmit Errors

Transmit errors detected by the GEMAC circuitry are noted in the transmit status word and with interrupts for maximum software implementation flexibility.

Receive Errors

The IEEE Std 802.3 prevents packets with error conditions from being passed to the software interface. The GEMAC circuitry does provide interrupts and status register bits to allow the software to monitor receive error conditions. When a received frame is detected to have an error, the data is discarded by using the packet FIFO mark and restore features. No receive status word or length is stored for that frame.

Design Constraints

The Ethernet Core requires design constraints to guarantee performance. These constraints should be placed in a .UCF file for the top level of the design. The following example of the constraint text is based on the port names of the Ethernet core. If these ports are mapped to FPGA pin names that are different, the FPGA pin names should be substituted for the port names in the example below.

TBD

Design Implementation

Design Tools

The GEMAC design is implemented using VHDL code and Xilinx Core Generator blocks.

Xilinx XST is the synthesis tool used for synthesizing the GEMAC. The NGC netlist output is then input to the Xilinx Foundation tool suite for device implementation.

Target Technology

The intended target technologies are Virtex™II or Virtex™II Pro FPGAs. Some features are not available in some target families.

Device Utilization and Performance Benchmarks

Since the GEMAC is a module that will be used with other design pieces in the FPGA, the utilization and timing numbers reported in this section are just estimates. As the GEMAC is combined with other pieces of the FPGA design, the utilization of FPGA resources and timing of the GEMAC design will vary from the results reported here.

In order to analyze the GEMAC's timing within the FPGA, a design will be created that instantiates the GEMAC with the following parameters set.

The GEMAC benchmarks are shown in Table 30 for a Virtex™II -6 FPGA.

Table 30: **GEMAC FPGA Performance and Resource Utilization Benchmarks (Virtex™II -6)**

Parameter Values					Device Resources				f_{MAX} (MHz)
					Slices	Slice Flip-Flops	BRAMS	4-input LUTs	

Notes:
1. F.26 with overall effort level of 5 and extra effort level of 2 into a xc2v6000-6-ff1152 device with opb_clk period constrained to 8 nS (125 Mhz)

The GEMAC benchmarks are shown in Table 31 for a Virtex™II Pro -7 FPGA

Table 31: **GEMAC FPGA Performance and Resource Utilization Benchmarks (Virtex™II Pro -7)**

Parameter Values					Device Resources				f_{MAX} (MHz)
					Slices	Slice Flip-Flops	BRAMS	4-input LUTs	

Notes:
1. F.26 with overall effort level of 5 and extra effort level of 2 into a xc2vp20-7-ff1152 device with opb_clk period constrained to 7 nS (142.8 Mhz)

Specification Exceptions

The GEMAC design currently has no exceptions to the mandatory IEEE Std. 802.3 MII interface requirements.

Reference Documents

The following document contains reference information important to understanding the GEMAC design:

- IEEE Std. 802.3

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/12/02	1.1	Initial Xilinx release
01/14/03	1.2	Update for EDP SP3
01/17/03	1.3	Update for release
01/21/03	1.4	Update PLB clk frequency requirements
05/13/03	1.5	Update for v1_00_b

Date	Version	Revision
05/15/03	1.6	Add auto negotiation bit to control register and add DMA registers
07/09/03	1.7	Update to new template
08/22/03	1.7.1	Minor edits per designer.