XPS Ethernet Lite Media Access Controller (v2.00a)

DS580 December 07, 2007 **Product Specification**

Introduction

The Ethernet Lite MAC (Media Access Controller) is designed to incorporate the applicable features described in the IEEE Std. 802.3 Media Independent Interface (MII) specification, which should be used as the definitive specification.

The Ethernet Lite MAC supports the IEEE Std. 802.3 Media Independent Interface (MII) to industry standard Physical Layer (PHY) devices and communicates to a processor via a Processor Local Bus (PLB) interface. The design provides a 10 Megabits per second (Mbps) and 100 Mbps (also known as Fast Ethernet) Interface. The goal is to provide the minimal functions necessary to provide an Ethernet interface with the least resources used.

Features

- PLB interface is based on PLB v4.6 specification
- Memory mapped direct I/O interface to the transmit and receive data dual port memory
- Media Independent Interface (MII) for connection to external 10/100 Mbps PHY transceivers
- Independent internal 2K byte Tx and Rx dual port memory for holding data for one packet
- Optional dual buffer memories, 4K byte ping-pong, for Tx and Rx
- Receive and Transmit Interrupts

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Spartan™-3, Spartan-3E, Spartan-3A, Spartan-3A DSP, Spartan-3AN, Virtex™-II P, Virtex-4, and Virtex-5	
Version of Core	xps_ethernetlite	v2.00a
Resources Used		
	Min	Max
Slices	Refer toTable 7, Table 8, and Table 9	
LUTs		
FFs		
Block RAMs		
Special Features	N/A	
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs & Application Notes	N/A	
Additional Items	N/A	
Design Tool Requirements		
Xilinx Implementation Tools	ISE 9.1i or later	
Verification	ModelSim SE/EE 6.0c or later	
Simulation	ModelSim SE/EE 6.0c or later	
Synthesis	XST 9.1i or later	
Support		
Provided by Xilinx, Inc.		

Functional Description

The XPS Ethernet Lite MAC is comprised of two blocks as shown in Figure 1. The PLB Slave Interface Module couple the Ethernet Lite MAC core to the PLB.

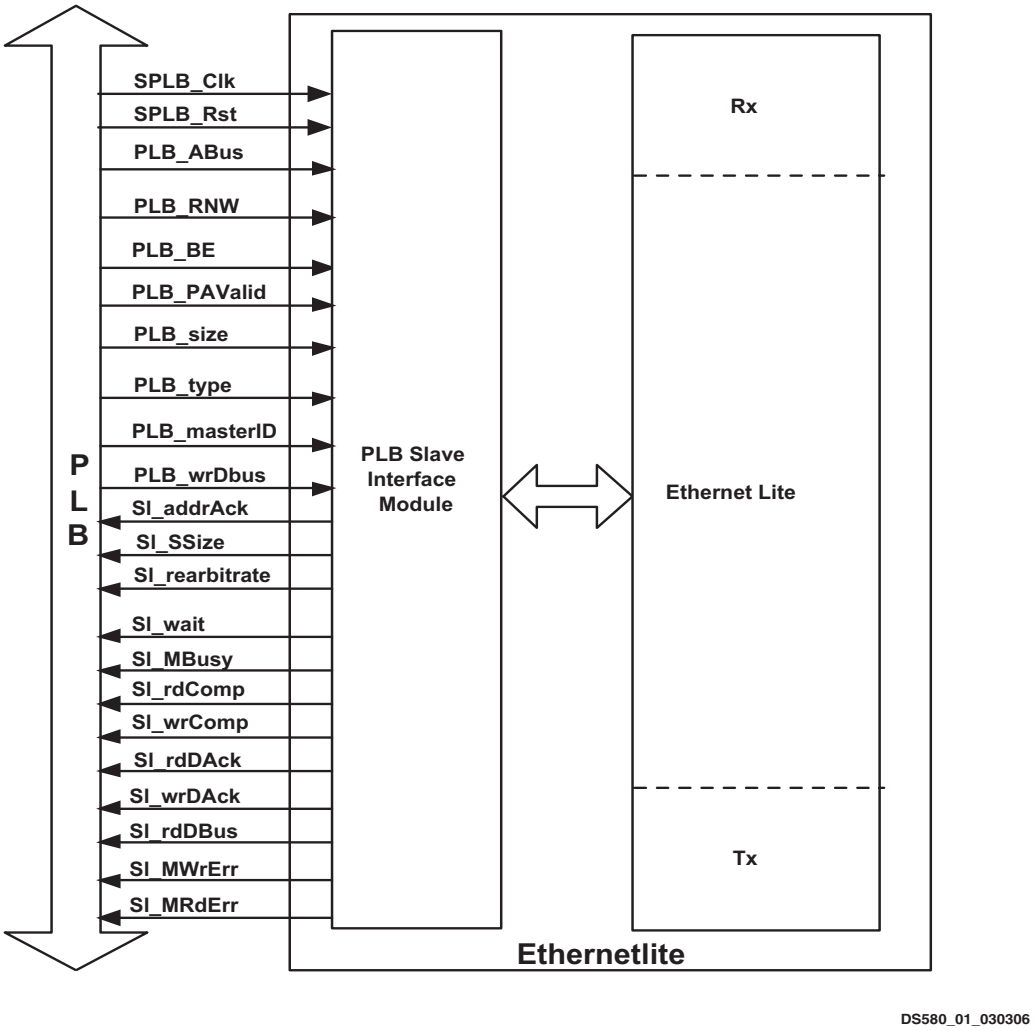

Figure 1: **XPS Ethernet Lite Block Diagram**

Ethernet Protocol

Ethernet data is encapsulated in frames as shown in Figure 2. 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. If a collision is detected during the transmission of the preamble or start of frame delimiter fields, the transmission of both fields will be completed.

For transmission, this field is always automatically inserted by the Ethernet Lite MAC and should never appear in the packet data provided to the Ethernet Lite MAC. For reception, this field is always stripped from the packet data. The Ethernet Lite MAC 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. If a collision is detected during the transmission of the preamble or start of frame delimiter fields, the transmission of both fields will be completed.

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 Ethernet Lite MAC and should never appear in the packet data provided to the Ethernet Lite MAC. 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 Ethernet Lite MAC supports transmission and reception of unicast and broadcast packets.

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.

Source Address

The source address field is 6 bytes in length[2]. 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.

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. Therefore, a value in this field that exceeds 1,500 (0x05DC) 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 (0x0600) or greater to signal a type field.

The Ethernet Lite MAC does not perform any processing of the type/length field.

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 Ethernet Lite MAC design does not support 16-bit destination addresses as defined in the IEEE 802 standard
2. The Ethernet Lite MAC 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. 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) which is required for successful CSMA/CD operation.

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 will be inserted automatically by the Ethernet Lite MAC if needed to meet the minimum length requirement. If present during receive packet, this field is always retained in the receive packet data.

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[1]:

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

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 Ethernet Lite MAC implementation of the CRC algorithm calculates the CRC value a nibble 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 is always inserted automatically by the Ethernet Lite MAC and is always retained in the receive packet data.

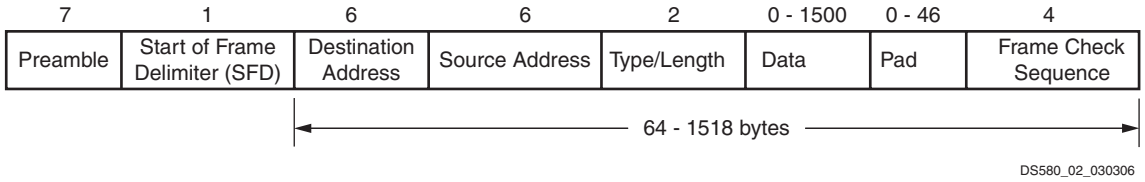

Figure 2: **Ethernet Data Frame**

Interframe Gap[2] 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 (9.6 uS for 10 MHz and 0.96 uS for 100 MHz). The process for deferring is different for half-duplex and full-duplex systems and is as follows:

1. Reference IEEE Std. 802.3 para. 3.2.8
2. Interframe Gap and interframe spacing are used interchangeably and are equivalent.

Half-Duplex

1. Even when it has nothing to transmit, the Ethernet Lite MAC monitors the bus for traffic by watching the carrier sense signal (CRS) from the external PHY. Whenever the bus is busy (CRS = '1'), the Ethernet Lite MAC defers to the passing frame by delaying any pending transmission of its own.
2. After the last bit of the passing frame (when carrier sense signal changes from true to false), the Ethernet Lite MAC starts the timing of the interframe gap.
3. The Ethernet Lite MAC will reset the interframe gap timer if carrier sense becomes true.

Full-Duplex

1. The Ethernet Lite MAC does not use the carrier sense signal from the external PHY when in full duplex mode because the bus is not shared and only needs to monitor its own transmissions. After the last bit of an Ethernet Lite MAC transmission, the Ethernet Lite MAC starts the timing of the interframe gap.

Carrier sense multiple access with collision detection (CSMA/CD) access method

A full duplex ethernet bus is by definition, a point to point dedicated connection between two ethernet devices capable of simultaneous transmit and receive with no possibility of collisions.

For a half duplex ethernet bus, the CSMA/CD media access method defines how two or more stations share a common bus.

To transmit, a station waits (defers) for a quiet period on the bus (no other station is transmitting (CRS = '0')) and then starts transmission of its message after the interframe gap period.

If, after initiating a transmission, the message collides with the message of another station (COL - '1'), then each transmitting station intentionally continues to transmit (jam) for an additional predefined period (32 bits for 10/100 Mbps) to ensure propagation of the collision throughout the system.

The station remains silent for a random amount of time (backoff) before attempting to transmit again.

A station can experience a collision during the beginning of its transmission (the collision window) before its transmission has had time to propagate to all stations on the bus. Once the collision window has passed, a transmitting station has acquired the bus.

Subsequent collisions (late collisions) are avoided because all other (properly functioning) stations are assumed to have detected the transmission and are deferring to it.

The time to acquire the bus is based on the round-trip propagation time of the bus (64 byte times for 10/100 Mbps).

Transmit Flow

The flow chart in Figure 3 shows the high level flow followed for packet transmission.

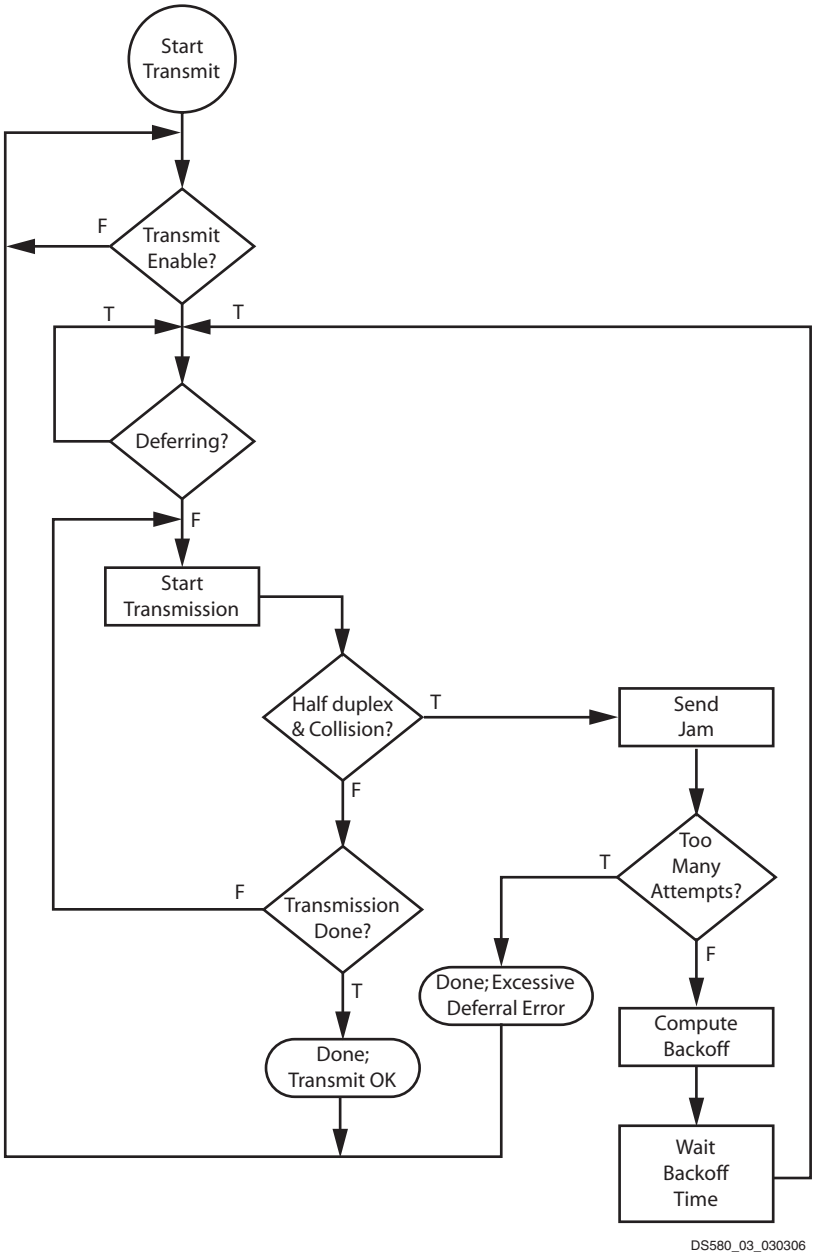

Figure 3: **Transmit Flow**

Receive Flow

The flow chart in Figure 4 shows the high level flow followed for packet reception.

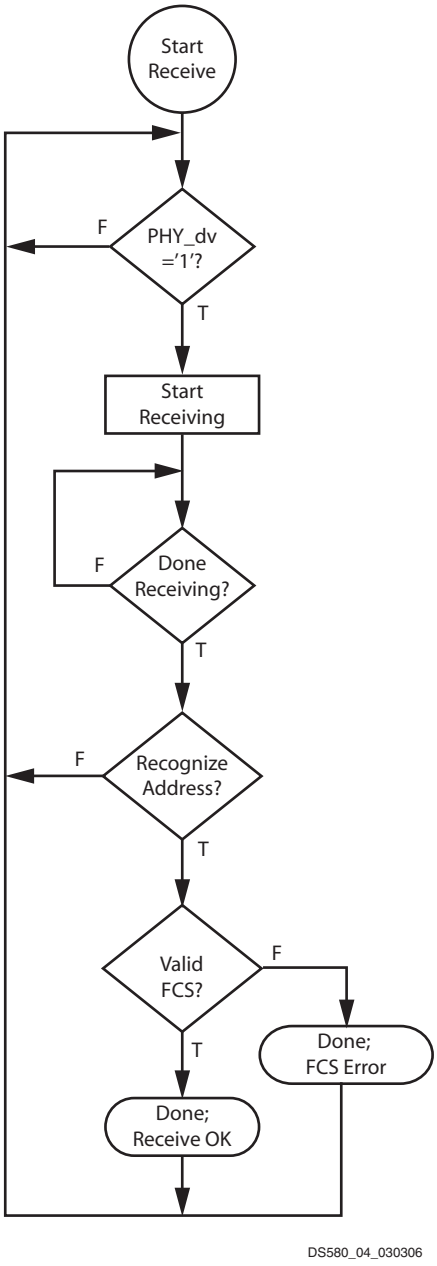

Figure 4: **Receive Flow**

Processor Interface

Single memory buffer Interface, C_TX_PING_PONG = C_RX_PING_PONG = 0

The Ethernet Lite MAC has a very simple interface to the processor without registers. The interface is implemented with a 32-bit wide data interface to a 4K byte block of dual port memory.

The dual port memory is allocated so that 2K bytes are dedicated to the transmit function and 2K bytes are dedicated to the receive function. This memory is capable of holding one maximum length ethernet packet in the receive and transmit memory areas simultaneously.

Transmit Interface

The transmit data should be stored in the dual port memory starting at address C_BASEADDR + 0x0. Due to the word aligned addressing, the second 4 bytes are located at C_BASEADDR + 0x4. The 32-bit interface requires that all 4 bytes be written at once, there is not individual byte enables within one 32- bit word.

The transmit data must include the destination address (6 bytes), the source address (6 bytes), the type/length field (2 bytes), and the data field (0 - 1500 bytes). The preamble, start of frame, and CRC should not be included in the dual port memory. The destination, source, type/length, and data must be packed together in contiguous memory.

Dual port memory address BASEADDR + 0x07F8 is used to set the global interrupt enable (GIE) bit. Setting the GIE = '0' prevents the IP2INTC_Irpt from going active during an interrupt event. Setting GIE = '1' allows the IP2INTC_Irpt to go active when an interrupt event has occurred.

Dual port memory addresses BASEADDR + 0x07F4 is used to store the length (in bytes) of the transmit data stored in dual port memory. The higher 8-bits of the length value should be stored in data bits (16 - 23), while the lower 8-bits should be stored data bits (24 - 31).

The least two significant bits of dual port memory address BASEADDR + 0x07FC are control bits (Program or "P" and Status or "S") that will be described below. The fourth bit (bit 28 on the data bus) (Transmit Interrupt Enable or "I") is used to enable transmit complete interrupt events. This event is a pulse and will occur anytime the memory is ready to accept new data. This includes the completion of programing the MAC address. The transmit complete interrupt will occur only if GIE and this bit are both set to '1'.

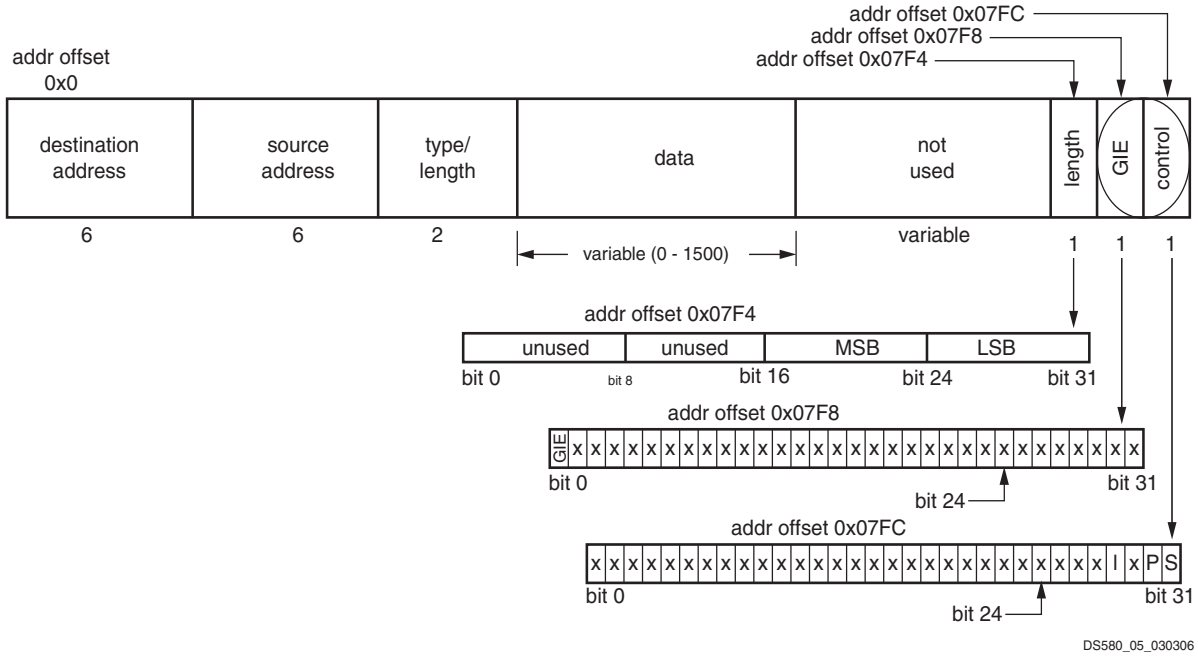

Figure 5: **Transmit Dual Port Memory**

The Ethernet Lite MAC requires that the length of the transmit data be stored in address offset 0x07F4 before the software sets the status bit at offset 0x07FC.

The proper software sequence for initiating a transmit is as follows:

- The software stores the transmit data in the dual port memory starting at address offset 0x0
- The software writes the length data in the dual port memory at address offset 0x07F4
- The software writes a '1' to the Status bit at address offset 0x07FC (bit 31 on the data bus)
- The software monitors the Status bit and waits until it is set to '0' by the Ethernet Lite MAC before initiating another transmit
- If the transmit interrupt and the global interrupt are both enabled, an interrupt will occur when the Ethernet Lite MAC clears the Status bit
- The transmit interrupt if enabled will also occur with the completion on writing the MAC address

Setting the Status bit to a '1' initiates the Ethernet Lite MAC transmit which will perform the following functions:

- Generate the preamble and start of frame fields
- Read the length and the specified amount of data out of the dual port memory according to the length value adding padding if required
- Detect any collision and performing any jamming, backoff, and retry if necessary
- Calculates the CRC and appends it to the end of the data
- Clears the status bit at the completion of the transmission
- Clearing the status bit will cause a transmit complete interrupt if enabled

Transmit Ping-Pong

If C_TX_PING_PONG is set to 1 then two memory buffers exist for the transmit data. The original (ping transmit buffer) remains at the same memory address and controls the global interrupt enable. The second (pong buffer) is mapped at BASEADDR + 0x0800 through 0x0FFC. The length and status must be used in the pong buffer the same as in the ping buffer. The I bit and GIE bit are not used from the pong buffer (i.e., the I bit and GIE bit of the ping buffer alone control the I bit and GIE bit settings for both buffers). The MAC address may be set from the pong buffer. The transmitter will always empty the ping buffer first after a reset. Then if data is ready to be transmitted from the pong buffer that will occur. However, if the pong buffer is not ready to transmit data the Ethernet Lite MAC will begin to monitor both the ping and pong buffer and transmit which ever buffer is ready first.

The proper software sequence for initiating a transmit with both a ping and pong buffer is as follows:

- The software stores the transmit data in the dual port memory starting at address offset 0x0
- The software writes the length data in the dual port memory at address offset 0x07F4
- The software writes a '1' to the Status bit at address offset 0x07FC (bit 31 on the data bus)
- The software may write to the pong buffer (0x0800 - 0x0FFC) at any time
- The software monitors the Status bit in the ping buffer and waits until it is set to '0', or waits for a transmit complete interrupt, before filling the ping buffer again
- If the transmit interrupt and the global interrupt are both enabled, an interrupt will occur when the Ethernet Lite MAC clears the Status bit
- The transmit interrupt if enabled will also occur with the completion of writing the MAC address

Setting the Status bit to a '1' initiates the Ethernet Lite MAC transmit which will perform the following functions:

- Generate the preamble and start of frame fields
- Read the length and the specified amount of data out of the dual port memory according to the length value adding padding if required
- Detect any collision and performing any jamming, backoff, and retry if necessary
- Calculates the CRC and appends it to the end of the data
- Clears the status bit at the completion of the transmission
- Clearing the status bit will cause a transmit complete interrupt if enabled
- The hardware will than transmit the pong buffer if it is available, or begin monitoring both ping and pong buffers until data is available

MAC Address

The 48-bit MAC address defaults at reset to 00-00-5E-00-FA-CE. This value can be changed by performing an address program operation via the transmit dual port memory.

The proper software sequence for programming a new MAC address is as follows:

- The software loads the new MAC address, in the transmit dual port memory starting at address offset 0x0. The most significant four bytes are stored at address offset 0x0 and the least significant two bytes are stored at address offset 0x4. The MAC address may also be programmed from the pong buffer starting at 0x0800
- The software writes a '1' to both the Program bit (bit 30 on the data bus) and the Status bit (bit 31 on the data bus) at address offset 0x07FC. The pong buffer address is 0x0FFC
- The software monitors the Status and Program bits and waits until they are set to '0's before performing any additional ethernet operations
- A transmit complete interrupt, if enabled, will occur when the Status and Program bits are cleared

Receive Interface

The entire received frame data from destination address to the end of the CRC is stored in the receive dual port memory area which starts at address BASEADDR + 0x1000. The preamble and start of frame fields are not stored in dual port memory.

Dual port memory address offset 0x17FC (bit 31 on the data bus) is used as a status to indicated the presence of a receive packet that is ready for processing by the software.

Dual port memory address offset 0x17FC (bit 28 on the data bus) is the Receive Interrupt enable. This event is a pulse and will occur anytime the memory has data available. The receive complete interrupt will occur only if GIE and this bit are both set to '1'.

When the Status bit is '0', the Ethernet Lite MAC will monitor the ethernet for packets with a destination address that matches its MAC address or the broadcast address. If a packet satisfies either of these conditions, the packet is received and stored in dual port memory starting at address offset 0x1000.

Once the packet has been received, the Ethernet Lite MAC verifies the CRC. If the CRC value is correct, the Status bit is set. If the CRC bit is incorrect, the status bit is not set and the Ethernet Lite MAC resumes monitoring the ethernet bus.

Once the Status bit is set, the Ethernet Lite MAC will not perform any receive operations until the bit has been cleared to '0' by software indicating that all of the receive data has been retrieved from the dual port memory.

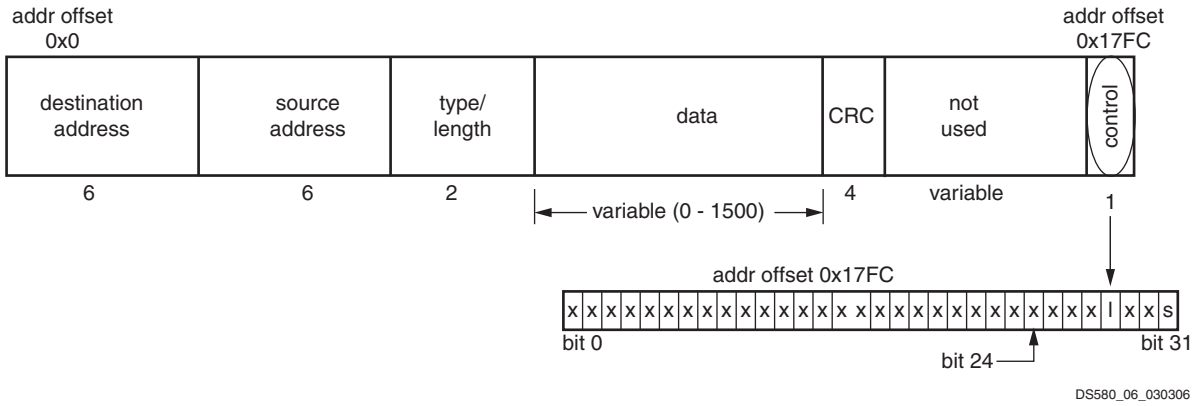

Figure 6: **Receive Dual Port Memory**

The proper software sequence for processing a receive is as follows:

- The software monitors the receive Status bit until it is set to '1' by the Ethernet Lite MAC, waits for a receive complete interrupt if enabled
- Once the Status is set to '1', or a receive complete interrupt has occurred, the software reads the entire receive data out of the dual port memory
- The software writes a '0' to the receive Status bit enabling the Ethernet Lite MAC to resume receive processing

Receive Ping-Pong

If C_RX_PING_PONG is set to '1' then two memory buffers exist for the receive data. The original (ping receive buffer) remains at the same memory location. The second (pong receiver buffer) is mapped at BASEADDR + 0x1800 through 0x1FFC. Data is stored the same in the pong buffer as it is in the ping buffer.

The proper software sequence for processing a receive packet(s) with C_RX_PING_PONG = 1 is as follows:

- The software monitors the ping receive Status bit until it is set to '1' by the Ethernet Lite MAC, or waits for a receive complete interrupt if enabled
- Once the ping Status is set to '1', or a receive complete interrupt has occurred, the software reads the entire receive data out of the ping dual port memory
- The Ethernet Lite MAC will receive the next packet and store it in the pong receive buffer
- The software writes a '0' to the ping receive Status bit enabling the Ethernet Lite MAC to receive another packet in the ping receive buffer
- The software monitors the pong receive Status bit until it is set to '1' by the Ethernet Lite MAC, or waits for a receive complete interrupt if enabled
- Once the pong Status is set to '1', or a receive complete interrupt has occurred, the software reads the entire receive data out of the ping dual port memory

- The hardware will always write the first received packet after a reset to the ping buffer, the second received packet will be written to the pong buffer and the third received packet will be written to the ping buffer

Table 1: **XPS Ethernet Lite MAC Memory Map**

Address Offset	Parameter Dependency	Memory Location Function
0x0000	Tx PING Buffer C_TX_PING_PONG = '0' or '1'	Destination Address Bytes 0 - 3 or MAC Address Bytes 0 - 3
0x0004		Destination Address Bytes 4 - 5 Source Address Bytes 0 - 1 or MAC Address Bytes 4 - 5
0x0008		Source Address Bytes 2 - 5
0x000C		Type/Length Field Data Field Bytes 0 - 1
0x0010 - 0x07F4		Remaining Data Field Bytes
0x07F4		Packet Length
0x07F8		Global Interrupt Enable
0x07FC		Control
0x0800	Tx PONG Buffer C_TX_PING_PONG = '1' else unused	Destination Address Bytes 0 - 3 or MAC Address Bytes 0 - 3
0x0804		Destination Address Bytes 4 - 5 Source Address Bytes 0 - 1 or MAC Address Bytes 4 - 5
0x0808		Source Address Bytes 2 - 5
0x080C		Type/Length Field Data Field Bytes 0 - 1
0x0810 - 0x0FF4		Remaining Data Field Bytes
0x0FF4		Packet Length
0x0FF8		Unused
0x0FFC		Control

Table 1: **XPS Ethernet Lite MAC Memory Map *(Contd)***

Address Offset	Parameter Dependency	Memory Location Function
0x1000	Rx PING Buffer C_RX_PING_PONG = '0' or '1'	Destination Address Bytes 0 - 3
0x1004		Destination Address Bytes 0 - 3 Source Address Bytes 4 - 5
0x1008		Source Address Bytes 2 - 5
0x100C		Type/Length Field Data Field Bytes 0 - 1
0x1010 - 0x17F4		Remaining Data and CRC Field Bytes
0x17FC		Control
0x1800	Rx PONG Buffer C_RX_PING_PONG = '1' else unused	Destination Address Bytes 0 - 3
0x1804		Destination Address Bytes 4 - 5 Source Address Bytes 0 - 1
0x1808		Source Address Bytes 2 - 5
0x180C		Type/Length Field Data Field Bytes 0 - 1
0x1810 - 0x1FF4		Remaining Data and CRC Field Bytes
0x1FFC		Control

XPS Ethernet Lite MAC I/O Signals

The XPS Ethernet Lite MAC I/O signals are listed and described in Table 2.

Table 2: **XPS Ethernet Lite MAC I/O Signal Description**

Port	Signal Name	Interface	I/O	Initial State	Description
System Signals					
P1	SPLB_Clk	PLB	I	-	PLB clock
P2	SPLB_Rst	PLB	I	-	PLB reset, active high
PLB Interface Signals					
P3	PLB_ABus[0:C_SPLB_AWIDTH - 1]	PLB	I	-	PLB address bus
P4	PLB_PAValid	PLB	I	-	PLB primary address valid
P5	PLB_masterID[0:C_SPLB_MID_WIDTH - 1]	PLB	I	-	PLB current master identifier
P6	PLB_RNW	PLB	I	-	PLB read not write
P7	PLB_BE[0:(C_SPLB_DWIDTH/8) - 1]	PLB	I	-	PLB byte enables
P8	PLB_size[0:3]	PLB	I	-	PLB size of requested transfer
P9	PLB_type[0:2]	PLB	I	-	PLB transfer type

Table 2: **XPS Ethernet Lite MAC I/O Signal Description *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P10	PLB_wrDBus[0:C_SPLB_DWIDTH - 1]	PLB	I	-	PLB write data bus
Unused PLB Interface Signals					
P11	PLB_UABus[0:C_SPLB_AWIDTH - 1]	PLB	I	-	PLB Upper Address bits
P12	PLB_SAValid	PLB	I	-	PLB secondary address valid
P13	PLB_rdPrim	PLB	I	-	PLB secondary to primary read request indicator
P14	PLB_wrPrim	PLB	I	-	PLB secondary to primary write request indicator
P15	PLB_abort	PLB	I	-	PLB abort bus request
P16	PLB_busLock	PLB	I	-	PLB bus lock
P17	PLB_MSize	PLB	I	-	PLB data bus width indicator
P18	PLB_lockErr	PLB	I	-	PLB lock error
P19	PLB_wrBurst	PLB	I	-	PLB burst write transfer
P20	PLB_rdBurst	PLB	I	-	PLB burst read transfer
P21	PLB_wrPendReq	PLB	I	-	PLB pending bus write request
P22	PLB_rdPendReq	PLB	I	-	PLB pending bus read request
P23	PLB_wrPendPri[0:1]	PLB	I	-	PLB pending write request priority
P24	PLB_rdPendPri[0:1]	PLB	I	-	PLB pending read request priority
P25	PLB_reqPri[0:1]	PLB	I	-	PLB current request priority
P26	PLB_TAttribute	PLB	I	-	PLB transfer attribute
PLB Slave Interface Signals					
P27	Sl_addrAck	PLB	O	0	Slave address acknowledge
P28	Sl_SSize[0:1]	PLB	O	0	Slave data bus size
P29	Sl_wait	PLB	O	0	Slave wait
P30	Sl_rearbitrate	PLB	O	0	Slave bus rearbitrate
P31	Sl_wrDAck	PLB	O	0	Slave write data acknowledge
P32	Sl_wrComp	PLB	O	0	Slave write transfer complete
P33	Sl_rdDBus[0:C_SPLB_DWIDTH - 1]	PLB	O	0	Slave read data bus
P34	Sl_rdDAck	PLB	O	0	Slave read data acknowledge
P35	Sl_rdComp	PLB	O	0	Slave read transfer complete
P36	Sl_MBusy[0:C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave busy

Table 2: **XPS Ethernet Lite MAC I/O Signal Description *(Contd)***

Port	Signal Name	Interface	I/O	Initial State	Description
P37	Sl_MWrErr[0:C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave write error
P38	Sl_MRdErr[0:C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Slave read error
Unused PLB Slave Interface Signals					
P39	Sl_wrBTerm	PLB	O	0	Slave terminate write burst transfer
P40	Sl_rdWdAddr[0:3]	PLB	O	0	Slave read word address
P41	Sl_rdBTerm	PLB	O	0	Slave terminate read burst transfer
P42	Sl_MIRQ[0:C_SPLB_NUM_MASTERS - 1]	PLB	O	0	Master interrupt request
Ethernet Lite MAC Signals					
P43	PHY_rx_data[3:0]	System	I	-	Ethernet receive data. Input from I/O block registers
P44	PHY_tx_data[3:0]	System	O	0	Ethernet transmit data. Output to I/O block registers
P45	PHY_dv	System	I	-	Ethernet receive data valid. Input from I/O block register
P46	PHY_rx_er	System	I	-	Ethernet receive error. Input from I/O block register
P47	PHY_tx_en	System	O	0	Ethernet transmit enable. Output to I/O block register
P48	PHY_tx_clk	System	I	-	Ethernet transmit clock input from input buffer
P49	PHY_rx_clk	System	I	-	Ethernet receive clock input from input buffer
P50	PHY_crs	System	I	-	Ethernet carrier sense input from input buffer
P51	PHY_col	System	I	-	Ethernet collision input from input buffer
P52	PHY_rst_n	System	O	1	Ethernet PHY reset output to output buffer
P53	IP2INTC_Irpt	System	O	0	System Interrupt

XPS Ethernet Lite MAC Design Parameters

To obtain an XPS Ethernet Lite MAC that is uniquely tailored to the user system requirements, certain features can be parameterized in the XPS Ethernet Lite MAC design. This allows a design that utilizes only the resources required by the system and runs at the best possible performance. The features that can be parameterized in the Xilinx XPS Ethernet Lite MAC design are shown in Table 3.

Table 3: **XPS Ethernet Lite MAC Design Parameters**

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
System parameters					
G1	Device family	C_FAMILY	“spartan3”, “spartan3e”, “spartan3a”, “spartan3an”, “virtex2p”,”virtex4”, “virtex5”	“virtex5”	string
PLB parameters					
G2	Device base address	C_BASEADDR	Valid Address range[1]	None[2, 3]	std_logic _ vector
G3	Device maximum address	C_HIGHADDR	Refer to *Allowable Parameter Combinations* section	None[2, 3]	std_logic _ vector
G4	BUS clock period in ps	C_SPLB_CLK_PERIOD_PS	Requirements as stated in note [4]	10000	integer
G5	PLB address bus width (in bits)	C_SPLB_AWIDTH	32	32	integer
G6	PLB data bus width (in bits)	C_SPLB_DWIDTH	32, 64, 128	32	integer
G7	Selects point-to-point or shared PLB topology	C_SPLB_P2P	0 = Shared bus topology	0	integer
G8	PLB Master ID width	C_SPLB_MID_WIDTH	$\log_2$ (C_SPLB_NUM_MASTERS), with minimum value of 1	1	integer
G9	Number of PLB Masters	C_SPLB_NUM_MASTERS	1 - 16	1	integer
G10	Width of IPIF Data Bus	C_SPLB_NATIVE_DWIDTH	32	32	integer
G11	Burst Support	C_SPLB_SUPPORT_BURSTS	0[5]	0	integer
Ethernet Lite MAC parameters					
G11	Half duplex transmit	C_DUPLEX	1 = Only full duplex operation available 0 = Only half duplex operation available	1	integer

Table 3: **XPS Ethernet Lite MAC Design Parameters *(Contd)***

Generic	Feature / Description	Parameter Name	Allowable Values	Default Value	VHDL Type
G12	Include second transmit buffer	C_TX_PING_PONG	1 = Two transmit buffers 0 = Single memory transmit buffer	0	integer
G13	Include second receive buffer	C_RX_PING_PONG	1 = Two receive buffers 0 = Single memory receive buffer	0	integer

Notes:

1. Address range specified by C_BASEADDR and C_HIGHADDR must be at least 0X2000 and must be a power of 2. C_BASEADDR must be a multiple of the range, where the range is C_HIGHADDR - C_BASEADDR + 1
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 1FFF
3. For example, C_BASEADDR = 0xE0000000, C_HIGHADDR = 0xE0001FFF
4. The PLB clock frequency must be $\geq$ 50 MHz for 100 Mbps ethernet operation and greater than or equal to 5 MHz for 10 Mbps ethernet operation
5. XPS Ethernetlite does not support bursts

Allowable Parameter Combinations

The XPS Ethernet Lite MAC is a synchronous design. Due to the state machine control architecture of receive and transmit operations, the PLB Clock must be greater than or equal to 50 MHz to allow ethernet operation at 100 Mbps and greater than or equal to 5 MHz for ethernet operation at 10 Mbps.

The address range specified by C_BASEADDR and C_HIGHADDR must be a power of 2, and C_HIGHADDR range must be at least 0x2000. For example, if C_BASEADDR = 0xE0000000, C_HIGHADDR must be at least = 0xE0001FFF.

XPS Ethernet Lite MAC Port Dependencies

The dependencies between the XPS Ethernet Lite MAC design parameters and I/O signals are described in Table 4. In addition, when certain features are parameterized out of the design, the related logic will no longer be apart of the design. The unused input signals and related output signals are set to a specified value.

Table 4: **XPS Ethernet Lite MAC Parameter - Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
Design Parameters				
G9	C_SPLB_NUM_MASTERS	P50, P51, P52, P52	-	Width of number of masters
G8	C_SPLB_MID_WIDTH	P20	-	Width of the PLB Address Bus
G5	C_SPLB_AWIDTH	P14	-	Width of the PLB Address Bus
G6	C_SPLB_DWIDTH	P24, P30, P45	-	Width of the PLB Data Bus and PLB Slave Data Bus

Table 4: **XPS Ethernet Lite MAC Parameter - Port Dependencies**

Generic or Port	Name	Affects	Depends	Relationship Description
I/O Signals				
P14	PLB_ABus[0:C_SPLB_AWIDTH - 1]	-	G5	Width of the PLB Address Bus varies according to C_SPLB_AWIDTH
P20	PLB_masterID[0:C_SPLB_MID_WIDTH - 1]	-	G8	Width of the PLB_mastedID varies according to C_SPLB_MID_WIDTH
P24	PLB_BE[0:(C_SPLB_DWIDTH/8) - 1]	-	G6	Width of the PLB Byte Enable varies according to C_SPLB_DWIDTH
P30	PLB_wrDBus[0:C_SPLB_DWIDTH - 1]	-	G6	Width of the PLB WriteData Bus varies according to C_SPLB_DWIDTH
P45	Sl_rdBus[0:C_SPLB_DWIDTH - 1]	-	G6	Width of the Slave Read Data Bus varies according to C_SPLB_DWIDTH
P50	Sl_MBusy[0:C_SPLB_NUM_MASTERS - 1]	-	G9	Width of the Sl_MBusy varies according to C_SPLB_NUM_MASTERS
P51	Sl_MWrErr[0:C_SPLB_NUM_MASTERS - 1]	-	G9	Width of the Sl_MWrErr varies according to C_SPLB_NUM_MASTERS
P52	Sl_MRdErr[0:C_SPLB_NUM_MASTERS - 1]	-	G9	Width of the Sl_MRdErr varies according to C_SPLB_NUM_MASTERS
P53	Sl_MIRQ[0:C_SPLB_NUM_MASTERS - 1]	-	G9	Width of the Sl_MIRQ varies according to C_SPLB_NUM_MASTERS

Clocks

The Ethernet Lite MAC design has three clock domains that are all asynchronous to each other. The clock domain diagram for the Ethernet Lite MAC is shown in Figure 7. These clock domains and any special requirements regarding them are discussed below.

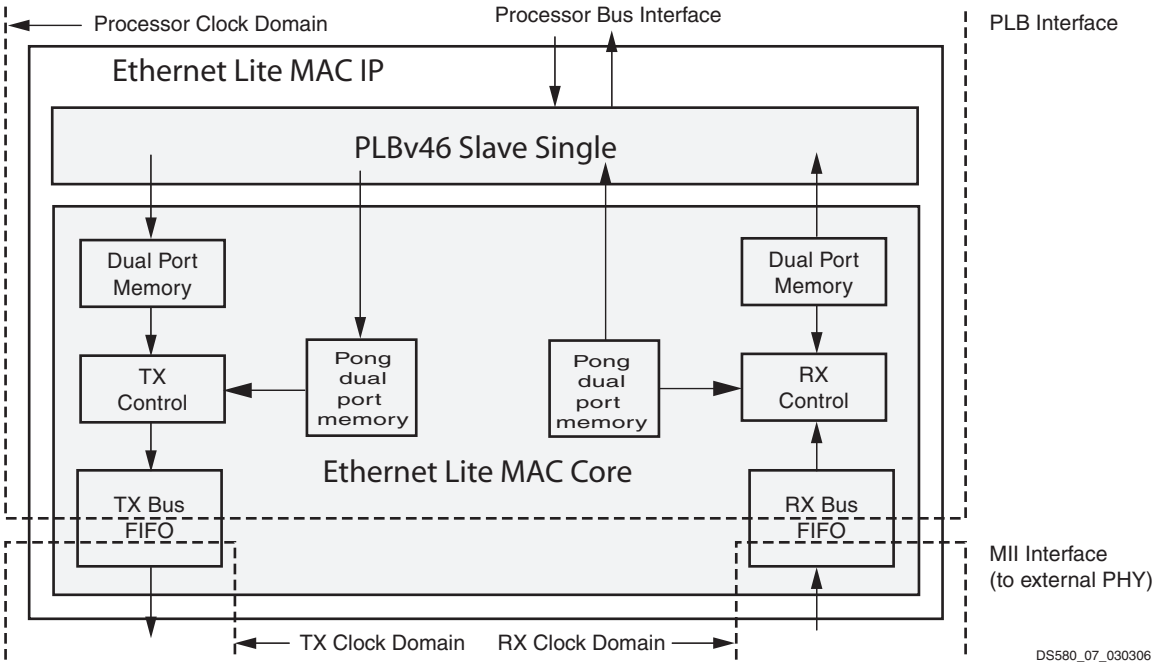

Figure 7: **Ethernet Lite MAC Clock Domain Diagram**

Transmit Clock

The transmit clock [TX_CLK] is generated by the external PHY and must be used by the Ethernet Lite MAC to provide transmit data [TXD (3:0)] and control signals [TX_EN and TX_ER] to the PHY.

The PHY provides one clock cycle for each nibble of data transferred resulting in a 2.5 MHz clock for 10BASE-T operation and 25 MHz for 100BASE-T operation at +/- 100 ppm with a duty cycle of between 35% and 65% inclusive. The PHY derives this clock from an external oscillator or crystal.

Receive Clock

The receive clock [RX_CLK] is also generated by the external PHY but is derived from the incoming ethernet traffic. Like the transmit clock, the PHY provides one clock cycle for each nibble of data transferred resulting in a 2.5 MHz clock for 10BASE-T operation and 25 MHz for 100BASE-T operation with a duty cycle of between 35% and 65% inclusive while incoming data is valid [RX_DV is '1'].

The minimum high and low times of the receive clock are at least 35% of the nominal period under all conditions. The receive clock is used by the Ethernet Lite MAC to sample the receive data [RXD(3:0)] and control signals [RX_DV and RX_ER] from the PHY.

Processor Bus Clock

The majority of the Ethernet Lite MAC operation functions in the processor bus clock domain. This clock must be greater than or equal to 50 MHz in order to transmit and receive ethernet data at 100 Mbps and greater than or equal to 5 MHz in order to transmit and receive ethernet data at 10 Mbps.

PHY Interface Signals

PHY_RST_N

Many PHY devices require that they be held in reset for some period after power becomes valid in order for the PHY device to be operational following the power-up sequence. The PHY_rst_n signal is an active low reset which is tied directly to the PLB reset signal (SPLB_Rst). This output signal may be connected to the active low reset input of a PHY device.

TX_EN

The Ethernet Lite MAC uses the Transmit Enable signal (TX_EN) to indicate to the PHY that it is providing nibbles at the MII interface for transmission. It is asserted synchronously to TX_CLK with the first nibble of the preamble and remains asserted while all nibbles have been transmitted. TX_EN is negated prior to the first TX_CLK following the final nibble of a frame.

This signal is transferred between the TX_CLK and processor clock domains at the asynchronous TX bus FIFO interface. The clock to output delay of this signal must be 0 to 25 nS. Figure 8 shows TX_EN timing during a transmission with no collisions.

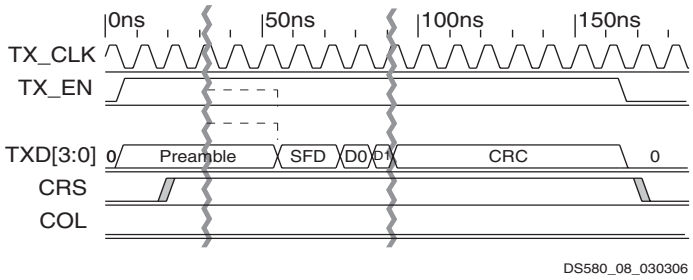

Figure 8: **Transmission with no Collision**

TXD(3:0)

The Ethernet Lite MAC drives the Transmit Data bus TXD(3:0) synchronously to TX_CLK. TXD(0) is the least significant bit. The PHY will transmit the value of TXD on every clock cycle that TX_EN is asserted.

This bus is transferred between the TX_CLK and processor clock domains at the asynchronous TX bus FIFO interface. The clock to output delay of this signal must be 0 to 25 nS. The order of the bits, nibbles, and bytes for transmit and receive are shown in Figure 9.

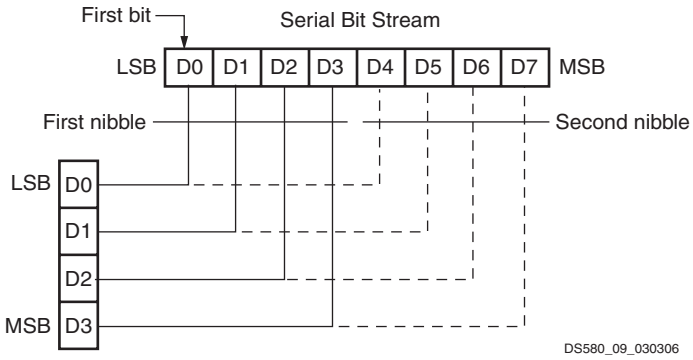

Figure 9: **Byte/Nibble Transmit and Receive Order**

RX_DV

The PHY drives the Receive Data Valid (RX_DV) signal to indicate that the PHY is driving recovered and decoded nibbles on the RXD(3:0) bus and that the data on RXD(3:0) is synchronous to RX_CLK. RX_DV is driven synchronously to RX_CLK. RX_DV remains asserted continuously from the first recovered nibble of the frame through the final recovered nibble and is negated prior to the first RX_CLK that follows the final nibble.

In order for a received frame to be correctly received by the Ethernet Lite MAC, RX_DV must encompass the frame, starting no later than the Start-of-Frame Delimiter (SFD) and excluding any End-of-Frame delimiter.

This signal is transferred between the RX_CLK and processor clock domains at the asynchronous RX bus FIFO interface. The PHY will provide a minimum of 10 nS setup and hold time for this signal in reference to RX_CLK. Figure 10 shows the behavior of RX_DV during frame reception.

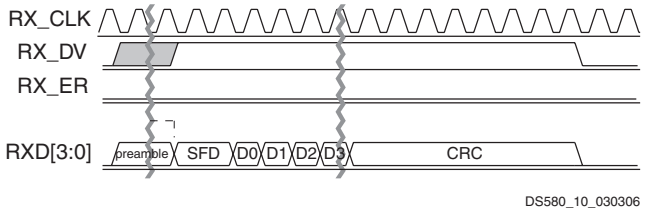

Figure 10: **Receive With No Errors**

RXD(3:0)

The PHY drives the Receive Data bus RXD(3:0) synchronously to RX_CLK. RXD(3:0) contains recovered data for each RX_CLK period in which RX_DV is asserted. RXD(0) is the least significant bit. The Ethernet Lite MAC must not be affected by RXD(3:0) while RX_DV is de-asserted.

Also, the Ethernet Lite MAC should ignore a special condition that occurs while RX_DV is de-asserted when the PHY may provide a False Carrier indication by asserting the RX_ER signal while driving the value 1110 onto RXD(3:0). This bus is transferred between the RX_CLK and processor clock domains at the asynchronous RX bus FIFO interface. The PHY will provide a minimum of 10 nS setup and hold time for this signal in reference to RX_CLK.

RX_ER

The PHY drives the Receive Error signal (RX_ER) synchronously to RX_CLK. The PHY drives RX_ER for one or more RX_CLK periods to indicate that an error (e.g., a coding error, or any error that the PHY is capable of detecting) was detected somewhere in the frame presently being transferred from the PHY to the Ethernet Lite MAC.

RX_ER should have no effect on the Ethernet Lite MAC while RX_DV is de-asserted. This signal is transferred between the RX_CLK and processor clock domains at the asynchronous RX bus FIFO interface.

The PHY will provide a minimum of 10 nS setup and hold time for this signal in reference to RX_CLK. Figure 11 shows the behavior of RX_ER during frame reception with errors.

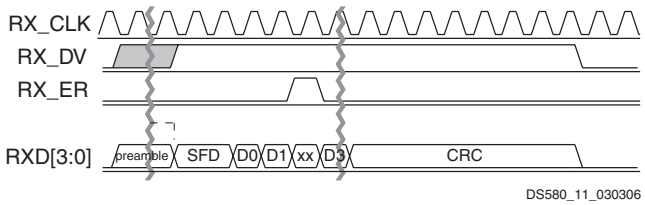

Figure 11: **Receive With Errors**

Table 5 shows the possible combinations for the receive signals.

Table 5: **Possible Values for RX_DV, RX_ER, and RXD[3:0]**

RX_DV	RX_ER	RXD[3:0]	Indication
0	0	0000 through 1111	Normal inter-frame
0	1	0000	Normal inter-frame
0	1	0001 through 1101	Reserved
0	1	1110	False carrier indication

CRS

The PHY drives the Carrier Sense signal (CRS) active to indicate that either the transmit or receive is non-idle when operating in half duplex mode. CRS is de-asserted when both the transmit and receive are idle.

The PHY drives CRS asserted throughout the duration of a collision condition. CRS is not synchronous to either the TX_CLK or the RX_CLK. The CRS signal is not used in full duplex mode. The CRS signal is used by both the Ethernet Lite MAC transmit and receive circuitry and is double synchronized to the processor clock as it enters the Ethernet Lite MAC.

COL

The PHY drives the Collision detected signal (COL) active to indicate the detection of a collision on the bus. The PHY drives CRS asserted while the collision condition persists. The PHY also drives COL asserted when operating at 10 Mbps for signal_quality_error (SQE) testing.

COL is not synchronous to either the TX_CLK or the RX_CLK. The COL signal is not used in full duplex mode. The COL signal is used by both the Ethernet Lite MAC transmit and receive circuitry and is double synchronized to the processor clock as it enters the Ethernet Lite MAC. Figure 12 shows the behavior of COL during frame transmission with a collision.

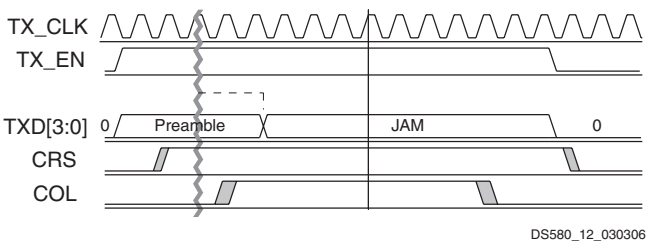

Figure 12: **Transmission With Collision**

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). The Ethernet Lite MAC accepts messages addressed to its unicast address and the broadcast address.

Design Constraints

The Ethernet Lite MAC 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 Lite MAC 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 shown in Figure 13.

```
NET "phy_rx_clk" TNM_NET = "phy_rx_clk";
NET "phy_tx_clk" TNM_NET = "phy_tx_clk";
OFFSET = OUT 10 ns AFTER "phy_tx_clk";
OFFSET = IN 6 ns BEFORE "phy_rx_clk";
NET "SPLB_Rst" TIG;
NET "phy_rx_clk" USELOWSKEWLINES;[1]
NET "phy_tx_clk" USELOWSKEWLINES;[1]
NET "phy_tx_clk" MAXSKEW = 2.0 ns;[2]
NET "phy_rx_clk" MAXSKEW = 2.0 ns;[2]
NET "phy_tx_clk" MAXSKEW = 1.0 ns;[3]
NET "phy_rx_clk" MAXSKEW = 1.0 ns;[3]
NET "phy_rx_clk" PERIOD = 40 ns HIGH 14 ns;
NET "phy_tx_clk" PERIOD = 40 ns HIGH 14 ns;
NET "phy_rx_data<3>" NODELAY;[1]
NET "phy_rx_data<2>" NODELAY;[1]
NET "phy_rx_data<1>" NODELAY;[1]
NET "phy_rx_data<0>" NODELAY;[1]
NET "phy_dv" NODELAY;[1]
NET "phy_rx_er" NODELAY;[1]
NET "phy_crs" NODELAY;[1]
NET "phy_col" NODELAY;[1]
NET "phy_rx_data<3>" IOBDELAY = NONE;[3]
NET "phy_rx_data<2>" IOBDELAY = NONE;[3]
NET "phy_rx_data<1>" IOBDELAY = NONE;[3]
NET "phy_rx_data<0>" IOBDELAY = NONE;[3]
NET "phy_dv" IOBDELAY = NONE;[3]
NET "phy_rx_er" IOBDELAY = NONE;[3]
NET "phy_crs" IOBDELAY = NONE;[3]
NET "phy_col" IOBDELAY = NONE;[3]

Notes: 1. Use for Spartan-3 devices only.
       2. Use for all devices except Virtex-4.
       3. Use for Virtex-4 only.
```

DS580_13_030306

Figure 13: **Design constraints**

Design Implementation

Target Technology

The intended target technology is Virtex-4, Virtex-5 and Spartan-3 family FPGAs.

Device Utilization and Performance Benchmarks

Core Performance

To analyze the XPS Ethernet Controller timing within the FPGA, a design was generated that enclosed the Core in a wrapper. For sizing estimates, a simple wrapper that connected all I/O to the ports of the wrapper was utilized. For Fmax, the wrapper was modified to incorporate input and output registers on all input and output ports respectively. FPGA performance and resource utilization benchmarks from the synthesis and mapping of the wrappers hosted in a xc5vlx110t-3 device are shown in Table 6.

Table 6: **XPS BRAM Interface Controller FPGA Performance and Resource Utilization Benchmarks**

Target FPGA			Device Resources			f_{MAX} (MHz)
C_SPLB_ SUPPORT_BURSTS	C_NATIVE_ DWIDTH	C_SPLB_ DWIDTH	Slices	Slice Flip-Flops	4-input LUTs	f_{MAX}
0	32	32		9	25	176
1	32	32		78	131	180
1	64	64		81	184	165
1	128	128		95	248	150

The XPS Ethernet Lite MAC resource utilization for various parameter combinations measured with Virtex-4 as the target device are detailed in Table 7.

Table 7: **Ethernet Lite MAC Performance and Resource Utilization for Virtex-4 (Device: xc4vlx40-ff1148-10)**

Parameter Values			Device Resources				Performance
C_DUPLEX	C_RX_PING_PONG	C_TX_PING_ PONG	Slices	Slice Flip-Flops	Block RAMS	LUTs	F_{Max} (MHz)
1	0	0	449	333	2	564	123
1	0	1	482	335	3	609	118
1	1	0	483	335	3	589	119
1	1	1	497	337	4	614	121
0	1	1	589	395	4	731	107

The XPS Ethernet Lite MAC resource utilization for various parameter combinations measured with Virtex-5 as the target device are detailed in Table 8

Table 8: **Ethernet Lite MAC Performance and Resource Utilization for Virtex-5 (Device: xc5vlx50-ff1153-1)**

Parameter Values			Device Resources			Performance
C_DUPLEX	C_RX_PING_PONG	C_TX_PING_ PONG	Slice Flip-Flops	LUTs	Block RAMS	F_{Max} (MHz)
1	0	0	332	466	2	123
1	0	1	334	485	3	121
1	1	0	334	476	3	142
1	1	1	336	524	4	150
0	1	1	394	627	4	125

The XPS Ethernet Lite MAC resource utilization for various parameter combinations measured with Spartan-3E as the target device are detailed in Table 9.

Table 9: **Ethernet Lite MAC Performance and Resource Utilization for Spartan-3E (Device: xc3s500e-fg320-5)**

Parameter Values			Device Resources				Performance
C_DUPLEX	C_RX_PING_PONG	C_TX_PING_ PONG	Slices	Slice Flip-Flops	Block RAMS	LUTs	F_{Max} (MHz)
1	0	0	361	333	2	525	109
1	0	1	380	335	3	615	106
1	1	0	497	335	3	592	103
1	1	1	456	337	4	592	103
0	1	1	491	395	4	705	102

System Performance

To measure the system performance (Fmax) of this core, this core was added to a Virtex-4 system, a Virtex-5 system, and a Spartan-3A system as the Device Under Test (DUT) as shown in Figure 14, Figure 15, and Figure 16.

Because the XPS Ethernet Lite MAC core will be used with other design modules in the FPGA, the utilization and timing numbers reported in this section are estimates only. When this core is combined with other designs in the system, the utilization of FPGA resources and timing of the design will vary from the results reported here.

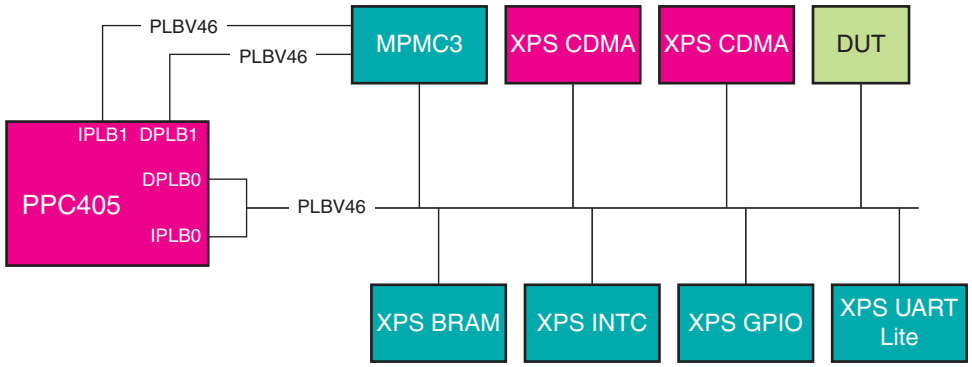

Figure 14: **Virtex-4 FX System**

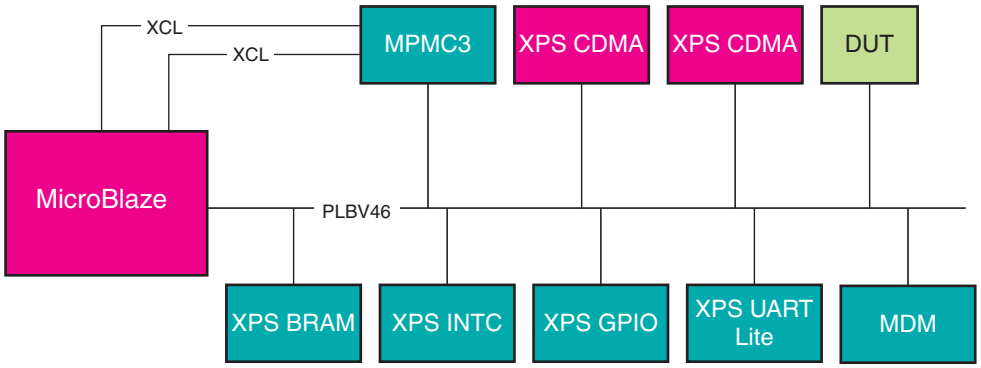

Figure 15: **Virtex-5 LX System**

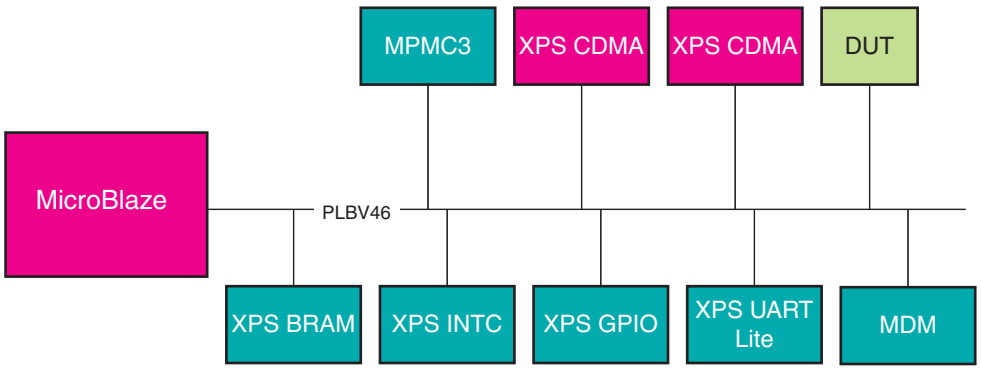

Figure 16: **Spartan-3A System**

The target FPGA was then filled with logic to drive the LUT and BRAM utilization to approximately 70% and the I/O utilization to approximately 80%. Using the default tool options and the slowest speed grade for the target FPGA, the resulting target FMax numbers are shown in Table 10.

Table 10: **XPS Ethernet Lite MAC System Performance**

Target FPGA	Target f_{MAX} (MHz)
S3A700 -4	90
V4FX60 -10	100
V5LXT50 -1	120

The target fMAX is influenced by the exact system and is provided for guidance. It is not a guaranteed value across all systems.

Reference Documents

The following document contains reference information important to understand the Ethernet Lite MAC design:

IBM CoreConnect 128-Bit Processor Local Bus: Architecture Specifications version 4.6.

Revision History

Date	Version	Revision
10/31/2006	1.0	Initial Xilinx release
9/26/07	1.1	Added FMax Margin System Performance section.
11/27/07	1.2	Added SP-3A DSP support.
07/12/07	1.3	New version of DS created to make the core license free