



10/100 and 1Gig Ethernet Media Access Controller

User's Guide

Introduction

This document serves as a guide containing technical information about the Lattice 10/100 and 1Gig Ethernet Media Access Controller (MAC) IP cores.

The 10/100 and 1Gig Ethernet MAC cores come with the following documentation and files:

- Data sheet
- Protected netlist/database
- Protected RTL simulation model
- Source files for instantiating the core

Core Specification

Features

- Compliant to IEEE 802.3z Standard
- Generic Host Interface
 - Configurable 8-bit or 16-bit Data Bus Widths
- 16-bit Wide Internal Data Path
- Generic Transmit and Receive FIFO Interface
- Full-duplex Operation in Gigabit Mode
- Full- and Half-duplex in 10/100 Mode
- Transmit and Receive Statistics Vector
- Programmable Inter Packet Gap (IPG)
- Multicast Address Filtering
- Supports
 - Full-duplex Control Using PAUSE Frames
 - VLAN Tagged Frames
 - Automatic Re-transmission on Collision
 - Automatic Padding of Short Frames
 - Multicast and Broadcast Frames
 - Optional FCS Transmission and Reception
 - Optional MII Management Interface Module
- Supports Jumbo Frames up to 8192 kbytes
- Reference Design for GMII to RGMII Bridge
 - Reduced pin count Gigabit Media Independent Interface

General Description

The Ethernet Media Access Controller (MAC) core can be configured to operate in either the Gigabit mode (1000 Mbits/sec data rate) or the Fast Ethernet mode (10/100 Mbits/sec data rate). Netlist configurations of this core operate only in either the Gigabit mode or Fast Ethernet mode. The netlist cannot auto-negotiate between the two different modes.

The Ethernet MAC transmits and receives data between a host processor and an Ethernet network. The main function of the Ethernet MAC is to ensure that the Media Access rules specified in the 802.3 IEEE standard are met while transmitting a frame of data over Ethernet. Figure 2 shows the transmission of data on the Ethernet network

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

The data received from the G/MII interface is first buffered until sufficient data is available to be processed by the Receive MAC (Rx MAC). The Preamble and the Start of Frame Delimiter (SFD) information are then extracted from the incoming frame to determine the start of a valid frame. The Receive MAC checks the address of the received packet and validates whether the frame can be received before transferring it onto the FIFO. Only valid frames are transferred into the FIFO. This feature has the following two benefits; the systems need not re-calculate the Frame Check Sequence (FCS) again when the frame is being transmitted, and it also keeps the receive MAC relatively simple. The Tri-Speed MAC however always calculates CRC to check whether the frame was received error-free or not.

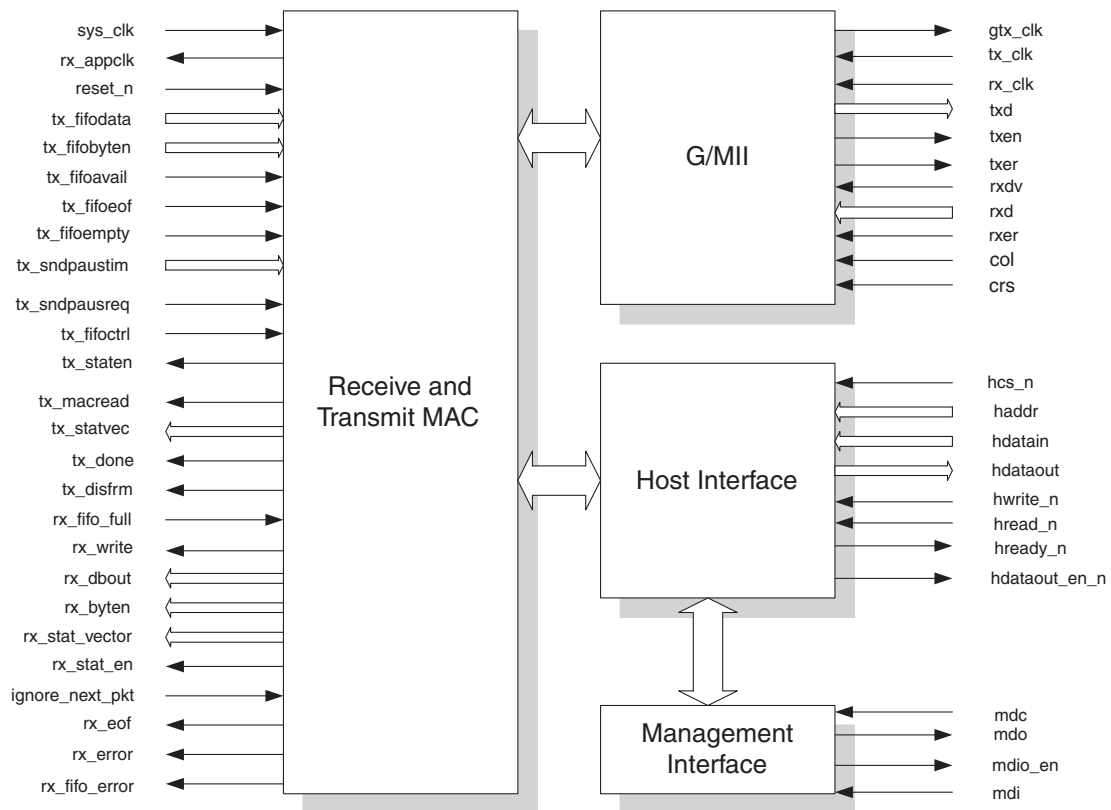
Figure 1. Un-Tagged Ethernet Frame Format

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

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

Block Diagram

Figure 2. 10/100 and 1Gig Ethernet MAC Block Diagram



Signal Descriptions

Table 1. Tri-Speed Ethernet MAC Input and Output Signals

| Port Name | Type | Active State | Description |
|---|--------|--------------|--|
| Clocks and Reset | | | |
| sys_clk | Input | N/A | System Clock. This is used to clock the host interface in all modes. In the Gigabit mode, the Tx MAC is clocked by this signal. All the input and the output signals of the Tx MAC are synchronous to this clock in the Gigabit mode. The frequency is always at 62.5 MHz. |
| rx_appclk | Output | N/A | Receive MAC Application I/F Clock. This clock is used in the Gigabit mode only. All the outputs driven by the Rx MAC are synchronous to this clock. The clock's frequency is 62.5 MHz. |
| gtx_clk | Output | N/A | Gigabit Transmit Clock. This clock is used in the Gigabit mode only. The transmit signals that are outputs on the GMII interface are synchronous to this clock. This clock has a frequency of 125 MHz. |
| tx_clk | Input | N/A | Transmit Clock. This clock is used in the 10/100 Mbps mode only. The Tx MAC, Tx MAC application interface and the MII are synchronous to this signal. This clock has a frequency of 2.5/25 MHz for 10/100 Mbps operation respectively. |
| rx_clk | Input | N/A | Receive Clock. This clock is an input from the PHY device. In the Gigabit mode, rx_clk frequency is 125 MHz. rx_clk is divided by two to provide the clock to the Receive MAC section. In the 10/100 mode, the corresponding rx_clk frequency is 2.5/25 MHz respectively, provided directly to the Receive MAC section. The receive signals at the GMII interface are always synchronous to rx_clk. |
| mdc | Input | N/A | Management Data Clock. This clock is used only when the Management Interface module is implemented. |
| reset_n | Input | Low | Reset. This is an active low signal that resets the internal registers and internal logic. When activated, the I/O signals are driven to their inactive levels. |
| Host Interface | | | |
| hcs_n | Input | Low | Chip Select. This is an active low signal used to select the core for register Read/Write operations. |
| haddr[7:0] | Input | N/A | Address. This selects one of the internal registers. |
| hdatain[(datawidth-1):0] | Input | N/A | Data Bus Input. The CPU writes to the internal registers through the data bus. |
| hwrite_n | Input | Low | Host Write. This active low signal is used to write data to the selected register. |
| hread_n | Input | Low | Host Read. This active low signal is used to read data from the selected register. |
| hready_n | Output | Low | Ready. This is an active low signal used to indicate the end of transfer. For write operations, hready_n is asserted after data is accepted (written). For read operations hready_n is asserted after data on the hdataout bus is ready to be driven out. |
| hdataout_en_n | Output | Low | Data Out Enable. This signal is driven low whenever the Tri-Speed MAC outputs valid data onto the hdataout bus. This signal can be used to build a bi-directional data bus. |
| hdataout[(datawidth-1):0] | Output | N/A | Data Bus Output. The CPU reads the internal registers through the data bus. |
| Transmit MAC Application Interface | | | |
| tx_fifodata[15:0] | Input | N/A | Transmit FIFO Read Data Bus. The data from the FIFO is presented on this bus. It is valid only when tx_fifobyten is non-zero. |

Table 1. Tri-Speed Ethernet MAC Input and Output Signals (Continued)

| Port Name | Type | Active State | Description |
|---------------------|--------|--------------|--|
| tx_fifobyten[1:0] | Input | N/A | Transmit FIFO Read Data Byte Enable. The upper bit validates the upper byte of the transmitted data. The lower bit validates the lower byte of the transmitted data. The Tri-Speed MAC expects packed data all the time except for the last word where only one byte could be valid. The Tri-Speed MAC assumes the natural end of frame when these bits do not have the value of 2'b11. |
| tx_fifoavail | Input | High | Transmit FIFO Data Available. When asserted, this indicates that the Tx FIFO has data ready for transmission. |
| tx_fifoeof | Input | High | Transmit FIFO End of Frame. This signal is asserted along with the last word/byte of frame data indicating the end of the frame. |
| tx_fifoempty | Input | High | Transmit FIFO Empty. This indicates that the Tx FIFO is empty. When this signal is asserted and the Tri-Speed MAC is reading the FIFO, the under-run condition is transferred to the network through the txer signal. |
| tx_sndpaustim[15:0] | Input | N/A | PAUSE Frame Timer. This indicates the PAUSE time value that should be sent in the PAUSE frame. |
| tx_sndpausreq | Input | High | PAUSE Frame Request. When asserted, the Tri-Speed MAC transmits a PAUSE frame. This is also the qualifying signal for the tx_sndpausetim bus. |
| tx_fifoctrl | Input | N/A | FIFO Control Frame. This signal indicates whether the current frame in the Transmit FIFO is a control frame or a data frame. It is qualified by the tx_avail signal. The following values apply: <ul style="list-style-type: none"> • 1 = Control frame • 0 = Normal frame |
| tx_staten | Output | High | Transmit Statistics Vector Enable. When asserted, the contents of the statistics vector bus tx_statvec are valid. |
| tx_macread | Output | High | Transmit FIFO Read. This is the Tri-Speed MAC Transmit FIFO read request, asserted by the Tri-Speed MAC for one clock only when it intends to read the FIFO. |
| tx_statvec[30:0] | Output | N/A | Transmit Statistics Vector. This includes useful information about the frame that was just transmitted. The corresponding bit locations of this bus are defined as follows: <ul style="list-style-type: none"> • tx_statvec[0] - UNICAST frame • tx_statvec[1] - Multicast frame • tx_statvec[2] - BROACAST frame • tx_statvec[3] - Bad FCS frame • tx_statvec[4] - JUMBO frame • tx_statvec[5] - FIFO under-run • tx_statvec[6] - PAUSE frame • tx_statvec[7] - VLAN tagged frame • tx_statvec[21:8] - Number of bytes in the transmitted frame • tx_statvec[22] - Deferred transmission • tx_statvec[23] - Excessive deferred transmission • tx_statvec[24] - Late collision • tx_statvec[25] - Excessive collision • tx_statvec[29:26] - Number of early collisions • tx_statvec[30] - FCS generation is disabled and a short frame was transmitted |
| tx_done | Output | High | Transmit Done. This signal is asserted for one clock cycle after transmitting a frame if no errors were present in transmission. |

Table 1. Tri-Speed Ethernet MAC Input and Output Signals (Continued)

| Port Name | Type | Active State | Description |
|--|--------|--------------|---|
| tx_discfrm | Output | High | Discard Frame. This signal is asserted at the end of a frame transmit process if the Tri-Speed MAC detected an error. The possible conditions are: <ul style="list-style-type: none"> • A FIFO under-run • Late collision (10/100 Mode only) • Excessive Collisions (10/100 Mode only) The user application normally moves the pointer to next frame in these conditions. |
| Management Interface Signals | | | |
| mdi | Input | High | Management Data Input. Used to transfer information from the PHY to the management module. |
| mdo | Output | High | Management Data Output. Used to transmit information from the management module to the PHY. |
| mdio_en | Output | High | Management Data Out Enable. Asserted whenever mdo is valid. This may be used to implement a bi-directional signal for mdi and mdo. |
| G/MII Signals | | | |
| txd[7:0] | Output | High | Transmit Data Sent to the PHY Chip. In Gigabit mode, txd[7:0] are used with a clock rate of 125 MHz. In 10/100 mode, only txd[3:0] are used with a clock rate of 2.5 MHz and 25 MHz respectively |
| txen | Output | High | Transmit Enable. Asserted by the Tri-Speed MAC to indicate the txd bus contains valid frame. |
| txer | Output | High | Transmit Error. Asserted when the Tri-Speed MAC generates a coding error on the byte currently being transferred. |
| rxdv | Input | High | Receive Data Valid. Indicates the data on the rxd bus is valid. |
| rxid[7:0] | Input | N/A | Receive Data Bus. Data is driven by the PHY on these lines, and is valid whenever rxdv is asserted. |
| rxer | Input | High | Receive Data Error. This signal is asserted by the external PHY device when it detects an error during frame reception. |
| col | Input | High | Collision. This active- high signal indicates a collision occurred during transmission. This signal is valid for half-duplex operation in Fast Ethernet (10/100) mode only. Otherwise, it is ignored. |
| crs | Input | High | Carrier Sense. This signal, when logic high, indicates the network has activity. Otherwise, it indicates the network is idle. This signal is valid for half-duplex operation in Fast Ethernet (10/100) mode only. |
| Receive MAC Application Interface | | | |
| rx_fifo_full | Input | High | Receive FIFO Full. This signal indicates the Rx FIFO is full and cannot accept any more data. This is an error condition and should never happen. |
| rx_write | Output | High | Receive FIFO Write. This signal is asserted by the Tri-Speed MAC core to request a FIFO write. |
| rx_dbout[15:0] | Output | N/A | Receive FIFO Data Output. This bus contains the data that is to be written into the Receive FIFO. |
| rx_byten[1:0] | Output | High | Receive FIFO Byte Enable. Indicates which of the bytes in the rx_dbout bus is valid. |
| rx_stat_vector[31:0] | Output | N/A | Receive Statistics Vector. This bus indicates the events encountered during frame reception. This bus is qualified by the rx_stat_en signal. The definition of each signal is explained in the Receive MAC section of this user's guide. |
| rx_stat_en | Output | High | Receive Statistics Vector Enable. When asserted, this indicates that the contents of the rx_stat_vector bus is valid. |

Table 1. Tri-Speed Ethernet MAC Input and Output Signals (Continued)

| Port Name | Type | Active State | Description |
|-----------------|--------|--------------|---|
| ignore_next_pkt | Input | High | Ignore Next Packet. This signal is asserted by the host to prevent a Receive FIFO Full condition. The Receive MAC continues dropping packets as long as this signal is asserted. This is an asynchronous signal. |
| rx_eof | Output | High | End Of Frame. Indicates all the data for the current packet has passed on to the FIFO. |
| rx_error | Output | High | Receive Packet Error. When asserted, this signal indicates the packet contains error(s). This signal is qualified with the rx_eof signal. |
| rx_fifo_error | Output | High | Receive FIFO Error. This signal is asserted when the rx_fifo_full signal was detected asserted during a FIFO write. It is qualified by rx_eof. |

Parameter Descriptions

The Tri-Speed MAC includes configurable parameters to allow easy integration with the user's application. The configurable parameters are shown in Table 2.

Table 2. Tri-Speed MAC Configuration Parameters

| Parameter | Value Range | Default | Description |
|----------------|-------------------------------------|----------------|---|
| MODE | 1000 Mbps Mode, or 10/100 Mbps Mode | 1000 Mbps Mode | This parameter defines the Ethernet speed the core will support. The terms "1000 Mbps Mode" and "Gigabit Mode" are used interchangeably in this document. "10/100 Mbps Mode" and "Fast Mode" are also used interchangeably. |
| MIIM_MODULE | Include, or Do Not Include | Include | This parameter determines whether or not the optional MIIM Module will be included in the core's implementation. |
| CPU_DATA_WIDTH | 8 bits, or 16 bits | 16 bits | This parameter determines the data bus width that will be used to communicate with the host. |

Functional Description

The Tri-Speed MAC is a fully synchronous machine composed of Transmit and Receive MAC sections that operate independently to support full duplex operation.

The block diagram of the Tri-Speed MAC core is shown in Figure 2. The major functional modules are:

- Host Interface
- Receive MAC
- Transmit MAC
- Internal Buffers and FIFO Interfaces
- G/MII
- (Optional) Management Interface Module

In the Gigabit mode, the 62.5 MHz system clock is supplied to the Transmit MAC. The system clock multiplied by two is used to clock the GMII interface for data transmission. When receiving data, an external PHY device provides the 125 MHz clock to the GMII receive section. The 125 MHz clock is divided by two and used to clock the Receive MAC.

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

Host Interface

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

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

Figure 9 shows the timing diagram associated with the host interface write and read operations.

Receive MAC (Rx MAC)

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

- Detect the Start of Frame
- Compare the MAC address
- Re-calculate CRC
- Process the Control Frame and pass it to the flow control module.

The Rx MAC operation is determined by programming the `MODE` and `TX_RX_CTL` registers.

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

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

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

The Rx MAC section operates on the `rx_clk` sourced from the PHY in the 10/100 mode. All the signals on the Receive MAC FIFO interface are synchronous to this clock. The Rx MAC operation is synchronized to an internal clock when the Tri-Speed MAC is in the Gigabit mode. This internal clock is a divided-by-two version of the `rx_clk` from the external PHY device. In this case, the divided clock is output to the FIFO interface as `rx_appclk`. The output signals from the Rx MAC interface are synchronous to this clock.

The Rx MAC is disabled while `reset_n` is low and should only be enabled after the associated registers are properly initialized.

Receiving Frames

The frames received by the Rx MAC are analyzed and the Preamble and SFD bytes are stripped off the frame before it is transferred to an external FIFO. The interface between the MAC and the FIFO is 16-bit wide. The byte that was received first is presented on the data bus lines [15:8], with bit[15] mapped to rxd[0] of the GMII bus. The data bus lines [7:0] carry the following byte, with bit [7] mapped to rxd[0].

The default behavior of the MAC is to transfer the unmodified frame after stripping off the Preamble and SFD bytes. This behavior can be changed by setting bit [1] of the TX_RX_CTL register. When bit [1] is set, the Rx MAC strips the Preamble, SFD, FCS bytes and the PAD bytes, if any.

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

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

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

Address Filtering

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

By default, the Rx MAC is configured to filter and discard Broadcast and Multicast frames. The MAC can be configured to receive Broadcast frames by setting bit [7] of the RX_TX_CTL register. Multicast frames are received only when bit [4] of the TX_RX_CTL register is set. When set, the Multicast frames are subject to filtering that is dependent on a hash table lookup. The six middle bits of the most significant byte of the CRC, calculated for the destination address field of the frame, are used to address one of the 64 bits of the hash table. If the retrieved bit is set, a Multicast addressed frame is received. If not, it is discarded.

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

Filtering based on Frame Length

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

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

Receiving a PAUSE Frame

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

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

Statistics Vector

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

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

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

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

Table 3. Receive Statistics Vector Description

| Bit | Description |
|------|--|
| 31 | Long Frame. This bit is set when a frame longer than specified in the MAX_FRAME_LENGTH register is received. |
| 30 | Short Frame. This bit is set when a frame shorter than the value specified in the MIN_FRAME_LENGTH register is received. |
| 29 | IPG Violation. This bit is set when a frame is received before the IPG timer runs out. |
| 28 | Preamble Shrink. This bit is asserted if the number of Preamble bytes received is not equal to seven. |
| 27 | Carrier Event Previously Seen. |
| 26 | Packet Ignored. When set, this indicates the incoming packet is to be ignored. |
| 25 | CRC Error. This bit is set when a frame is received with an error in the CRC field. |
| 24 | Length Check Error. This bit is set if the number of data bytes in the incoming frame matches the value in the length field of the frame. |
| 23 | Receive OK. This bit is set if the frame is received without any error. |
| 22 | Multicast Address. This bit is set to indicate the received frame contains a Multicast Address. |
| 21 | Broadcast Address. This bit is set to indicate the received frame contains a Broadcast Address. |
| 20 | Dribble Nibble. This bit is set when only 4 bits of the data presented on the RS interface are valid. |
| 19 | Unsupported Opcode. This bit is set if the received control frame has an unsupported opcode. In this version of the IP, only the opcode for PAUSE frame is supported. |
| 18 | Control Frame. This bit is set to indicate that a Control frame was received. |
| 17 | PAUSE Frame. This bit is set when the received Control frame contains a valid PAUSE opcode. |
| 16 | VLAN Tag Detected. This bit is set when the Tri-Speed MAC receives a VLAN Tagged frame. |
| 15:0 | Frame Byte Count. This contains the length of the frame that was received. The frame length includes the DA, SA, L/T, TAG, DATA, PAD and FCS fields. |

Transmit MAC (Tx MAC)

The Tx MAC is responsible for controlling access to the physical medium. The TxMAC reads data from an external TxFIFO when it detects an active `tx_fifoavail`. The Tx MAC then formats this data into an Ethernet packet and passes it to the G/MII module.

The Tx MAC is disabled while `reset_n` is low and should only be enabled only after the associated registers are properly initialized. Once enabled, the Tx MAC will continuously monitor the FIFO interface for an indication that frame(s) are ready to be transmitted. In the Gigabit mode, Tx MAC and the TxFIFO interface operations are synchronous to `sys_clk`. In the 10/100 mode, the Tx MAC is clocked by `tx_clk` (supplied from the PHY device). The TxFIFO interface signals in the 10/100 mode are synchronous to `tx_clk`.

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

Transmitting Frames

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

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

The input signal `tx_eof` is asserted along with the last set of data transfer to indicate the end of the frame. The Tx MAC requires a continuous stream of data for the entire frame. There cannot be any bubbles of "no data transfer" within a frame. The only exception to this rule is the transfer of last set of data which can have only one byte enabled. If the MAC is able to transmit the frame without any errors, the `tx_done` signal is asserted. Once the transmission has ended, data on the `tx_stat_vector` bus is presented to the host - including all the statistical information collected in the process of transmitting the frame. Data on this bus is qualified by assertion of the `tx_staten` signal.

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

External Transmit FIFO

The interface between the Tx MAC and the FIFO is 16-bit wide. The byte to be transmitted first is presented in position 15:8. The byte presented on bits 7:0 will be transmitted next. Within the respective bytes, the bit presented on positions 15(7) is transmitted first and the bit in position 8(0) is transmitted last. In other words, bit[15]([7]) will be transmitted on the `txd[0]` signal of GMII while the bit[8]([0]) will be transmitted on `txd[7]`. Byte Enable bit[1] corresponds to the byte in position 15:8 and byte enable [0] corresponds to the byte [7:0].

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

FIFO Under-flow

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

Transmitting PAUSE Frame

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

Retries on Collision

When operating in the half-duplex mode, the Transmit MAC has the capability to perform re-transmission of frames that have experienced in-window collision up to the specified maximum. This is possible because the MAC always buffers the first 64 bytes of the frame. This feature can be disabled by setting bit[6] of the `TX_RX_CTL` register. When retries are disabled, it is the application software's responsibility to perform retries of collided packets.

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

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

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

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

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

Internal Data Buffer and FIFO Interfaces

In the 10/100 mode, the Transmit and Receive sections each contain FIFOs to handle packets less than 64 bytes and to provide additional data buffering for normal packets. External Transmit and Receive FIFOs are required to store variable-length normal packets.

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

The core provides a feature where the user can block all the frames that are shorter than the minimum frame length of 64 bytes in the Tri-speed MAC itself. This prevents the collision fragments from reaching the user's application. The Receive Section contains an internal buffer to support this feature.

In the Gigabit mode, only the Receive Section includes a FIFO and this support the same functions as in the 10/100 mode. External Transmit and Receive FIFOs are required to store variable-length normal packets.

The Tri-Speed MAC provides two independent interfaces for use with external Transmit and Receive FIFOs. This feature enables the Tri-Speed MAC to support full duplex operation in either 10/100 or Gigabit mode.

G/MII Interface

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

The module implementing the interface also converts the data to a format usable by the MAC. In the Gigabit mode, the 8-bit data at the interface is converted to the 16-bit data path of the MAC. In the 10/100 mode, the 4-bit MII data is packed and input to the 16-bit MAC.

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

(Optional) Media Independent Interface Management Module (MIIM)

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

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

Internal Registers

The Tri-Speed MAC internal registers are initialized through the generic Host Interface. These rules apply when accessing the internal registers:

- In 8-bit Host Interfaces, the individual bytes of the registers are accessed through their corresponding addresses, with the lower address pointing to the lower byte.
- In 16-bit or greater Host Interfaces, only even numbered addresses should be used.
- The reserved bits should be programmed to 0. These bits are invalid, and should be discarded when read.
- All registers except the MODE register can be written into only when the core is in the IDLE state. The MODE register is the only register that can be written after the Tri-Speed MAC is no longer in the Reset condition.

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

Table 4. Tri-Speed MAC Internal Registers

| Register Description | Mnemonic | I/O Address | POR Value |
|--|--------------|-------------|-----------|
| Mode register | MODE | 00H - 01H | 0000H |
| Transmit and Receive Control register | TX_RX_CTL | 02H - 03H | 0000H |
| Maximum Packet Size register | MAX_PKT_SIZE | 04H - 05H | 05EEH |
| Inter Packet Gap register | IPG_VAL | 08H - 09H | 0048H |
| Tri-Speed MAC Address register 0 | MAC_ADDR_0 | 0AH - 0BH | 0000H |
| Tri-Speed MAC Address register 1 | MAC_ADDR_1 | 0CH - 0DH | 0000H |
| Tri-Speed MAC Address register 2 | MAC_ADDR_2 | 0EH - 0FH | 0000H |
| Transmit and Receive Status | TX_RX_STS | 12H - 13H | 0000H |
| GMII Management Interface Control register | GMII_MNG_CTL | 14H - 15H | 0000H |
| GMII Management Data register | GMII_MNG_DAT | 16H - 17H | 0000H |
| VLAN Tag Length/type register | VLAN_TAG | 32H - 33H | 0000H |
| Multicast_table_0 | MLT_TAB_0 | 22H - 23H | 0000H |
| Multicast_table_1 | MLT_TAB_1 | 24H - 25H | 0000H |
| Multicast_table_2 | MLT_TAB_2 | 26H - 27H | 0000H |
| Multicast_table_3 | MLT_TAB_3 | 28H - 29H | 0000H |
| Multicast_table_4 | MLT_TAB_4 | 2AH - 2BH | 0000H |
| Multicast_table_5 | MLT_TAB_5 | 2CH - 2DH | 0000H |
| Multicast_table_6 | MLT_TAB_6 | 2EH - 2FH | 0000H |
| Multicast_table_7 | MLT_TAB_7 | 30H - 31H | 0000H |
| Pause_opcode | PAUS_OP | 34H - 35H | 0080H |

Register Descriptions

MODE (R/W)

Mnemonic: MODE

POR Value = 0000H

| Name | Range | Description |
|---------|-------|--|
| Rsvd | 15:4 | Reserved. |
| Tx_en | 3 | Transmit Enable. When this bit is set, the Tx MAC is enabled to transmit frames. When reset, the Tx MAC completes transmission of the packet currently being processed, then stops. |
| Rx_en | 2 | Receive Enable. When this bit is set, the Rx MAC is enabled to receive frames. When reset, the Rx MAC completes reception of the packet currently being processed, then stops. |
| FC_en | 1 | Flow-control Enable. When set, this enables the flow control functionality of the Tx MAC. This bit should be set for the Tx MAC either to pause or to transmit a PAUSE frame. |
| Gbit_en | 0 | Gigabit Enable. In Gigabit mode, this bit is always high and cannot be overwritten. In 10/100 mode, this bit is always low and cannot be overwritten. |

Transmit and Receive Control (R/W)

Mnemonic: TX_RX_CTL

POR Value = 0000H

This register can be overwritten only when the Rx MAC and the Tx MAC are disabled. This register controls the various features of the MAC.

| Name | Range | Description |
|----------------|-------|---|
| Rsvd | 15:9 | Reserved. |
| Receive_short | 8 | Receive Short Frames. When high, enables the Rx MAC to receive frames shorter than 64 bytes. |
| Receive_brdcst | 7 | Receive Broadcast. When high, enables the Rx MAC to receive broadcast frames |
| Dis_rtry | 6 | Disable Retry (10/100 mode only). When high, disables retry on collision. |
| Hden | 5 | Half-duplex Enable (10/100 mode only). When high, configures the Tx MAC to operate in half-duplex mode. |
| Receive_mltcst | 4 | Receive Multicast. When high, the multicast frames will be received per the filtering rules for such frames. When low, no Multicast (except PAUSE) frames will be received. |
| Receive_pause | 3 | Receive PAUSE. When set, the Rx MAC will indicate the PAUSE frame reception to the Tx MAC. In either case, PAUSE frames are received and transferred to the FIFO. |
| Tx_dis_fcs | 2 | Transmit Disable FCS. When set, the FCS field generation is disabled in the Tx MAC. |
| Discard_fcs | 1 | Rx Discard FCS and Pad. When set, the FCS and any of the padding bytes are stripped off the frame before it is transferred to the FIFO. When low, the entire frame is transferred as is. |
| Prms | 0 | Promiscuous Mode. When asserted, all filtering schemes are abandoned and the Rx MAC receives frames with any address. |

Maximum Packet Size (R/W)

Mnemonic: MAX_PKT_SIZE

POR Value = 05EEH (1518 decimal)

This register can be overwritten only when the MAC is disabled. All frames longer than the value (number of bytes) in this register will be tagged as long frames.

| Name | Range | Description |
|-----------|-------|---|
| Max_frame | 15:0 | Maximum size of the packet than can be handled by the core. |

IPG (Inter Packet Gap) (R/W)

Mnemonic: IPG_VAL

POR Value = 0048H

| Name | Range | Description |
|------|-------|--|
| Rsvd | 15:5 | Reserved. |
| IPG | 4:0 | Inter-packet gap value in units of bit time. |

MAC Address Register {0,1,2} (R/W), Set of Three

Mnemonic: MAC_ADD

POR Value = 0000H.

The MAC Address Registers 0-2 contain the Ethernet address of the port. The MAC Address Register [0] has the two bytes that are transmitted first and the MAC Address Register [2] has the two bytes that are transmitted last. Bit[15] is transmitted first while bit[0] is transmitted last.

| Name | Range | Description |
|----------|-------|---|
| Mac_addr | 15:0 | Ethernet address assigned to the port supported by the Tri-speed MAC. |

Transmit and Receive Status (RO)

Mnemonic: TX_RX_STS

POR Value = 0000H

This register reports events that have occurred during packet reception and transmission.

| Name | Range | Description |
|---------------|-------|---|
| Rsvd | 15:11 | Reserved. |
| Rx_idle | 10 | Receive MAC Idle. Receive MAC in idle condition used to reset configurations by CPU interface. |
| Tagged_frame | 9 | Tagged Frame. Tagged frame received. |
| Brdcst_frame | 8 | Broadcast Frame. Indicates that a Broadcast packet was received. |
| Multcst_frame | 7 | Multicast Frame. Indicates that a Multicast packet was received. |
| IPG_shrink | 6 | IPG Shrink. Received frame with shrunk IPG (IPG < 96 bit time). |
| Short_frame | 5 | Short Packet. Indicates that a packet shorter than 64 bytes has been received. |
| Long_frame | 4 | Too Long Packet. Indicates receipt of a packet longer than the maximum allowable packet size specified in the MAX_PKT_SIZE register. |
| Error_frame | 3 | Rx_er Asserted. Indicates the frame was received with the rx_er signal asserted. |
| CRC | 2 | CRC Error. Indicates a packet was received with a CRC error. |
| Pause_frame | 1 | PAUSE Frame. Indicates a PAUSE frame was received. |
| Tx_idle | 0 | Transmit MAC Idle. Transmit MAC in idle condition, used to reset configurations by CPU interface. |

VLAN Tag (RO)

Mnemonic: VLAN_TAG

POR Value = 0000H.

The VLAN tag register has the VLAN TAG field of the most recent tagged frame that was received. This is a read only register.

| Name | Range | Description |
|------|-------|--|
| VLAN | 15:0 | This field defines length/type of field of the VLAN tag when inserted into transmitted frames. |

GMII Management Register Access Control (R/W)

Mnemonic: GMII_MNG_CTL

POR Value = 0000H.

The GMII Management Access register controls the Management Interface Module. This register can be overwritten only when the interface is not busy. A write operation will be ignored when the interface is busy.

| Name | Range | Description |
|-----------|-------|--|
| Rsvd | 15 | Reserved. |
| Cmd_fin | 14 | Command Finished. When high, it means the interface has completed the intended operation. This bit is set to 0 when the interface is busy. |
| RW_phyreg | 13 | Read/Write PHY Registers <ul style="list-style-type: none"> • When '1' -> write operation • When '0' -> read operation |
| Phy_addr | 12:8 | GMII PHY Address. The address of the accessed PHY Bit 12 is the most significant bit, and it is the first PHY address bit to be transmitted and received. |
| Rsvd | 7:5 | Reserved. |
| Reg_addr | 4:0 | GMII Register Address. The address of the register accessed. Bit 4 is the most significant bit and is the first register address bit to be transmitted or received. |

GMII Management Access Data (R/W)

Mnemonic: GMII_MNG_DAT

POR Value = 0000H.

The contents of this register will be transmitted when a write operation is to be performed. When a read operation is performed, this register will contain the value that was read from a PHY register. This register should be read only after the cmd_fin bit in the control register is set.

| Name | Range | Description |
|----------|-------|---|
| GMII_dat | 15:0 | GMII Data. Bit 15 is the most significant bit, corresponding to bit 15 of the accessed register. |

Multicast Tables (R/W), set of eight

Mnemonic: MLT_TAB_[0-7]

POR Value = 0000H.

When the core is programmed to receive multicast frames, a filtering scheme is used to decide whether the frame should be received or not. The six middle bits of the most significant byte of the CRC value, calculated for the destination address, are used as a key to the 64-bit hash table. The three most significant bits select one of the eight tables, and the three least significant bits select a bit. The frame is received only if this bit is set.

| Name | Range | Description |
|-----------------------|-------|---|
| Multicast_table_[0-7] | 15:0 | Multicast Table. Eight tables that make a 64-bit hash. |

Pause Opcode (R/W)

Mnemonic: PAUS_OP
POR Value = 0080H

This register contains the PAUSE Opcode, This will be compared against the Opcode in the received PAUSE frame. This value will also be included in any PAUSE frame transmitted by the Tri-Speed MAC. Bit 15 is transmitted first and bit 0 is transmitted last.

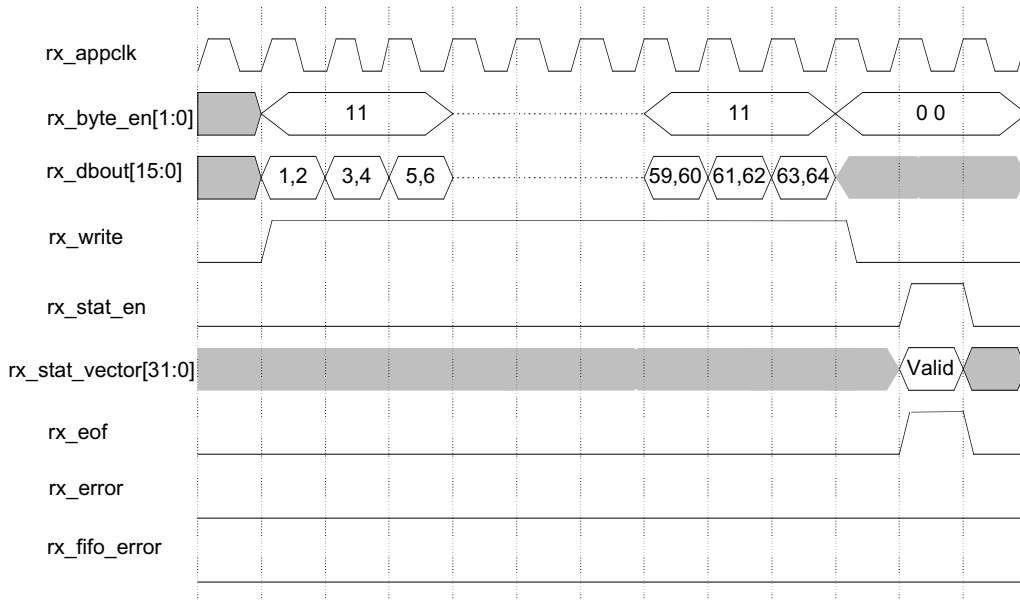
| Name | Range | Description |
|--------------|-------|---------------|
| Pause_OpCode | 15:0 | PAUSE Opcode. |

Timing Diagrams

The operational timing diagrams applicable to the Tri-speed MAC interfaces are shown below:

1. Reception of a 64-Byte Frame Without Error -Rx MAC Application Interface (Gigabit Mode)

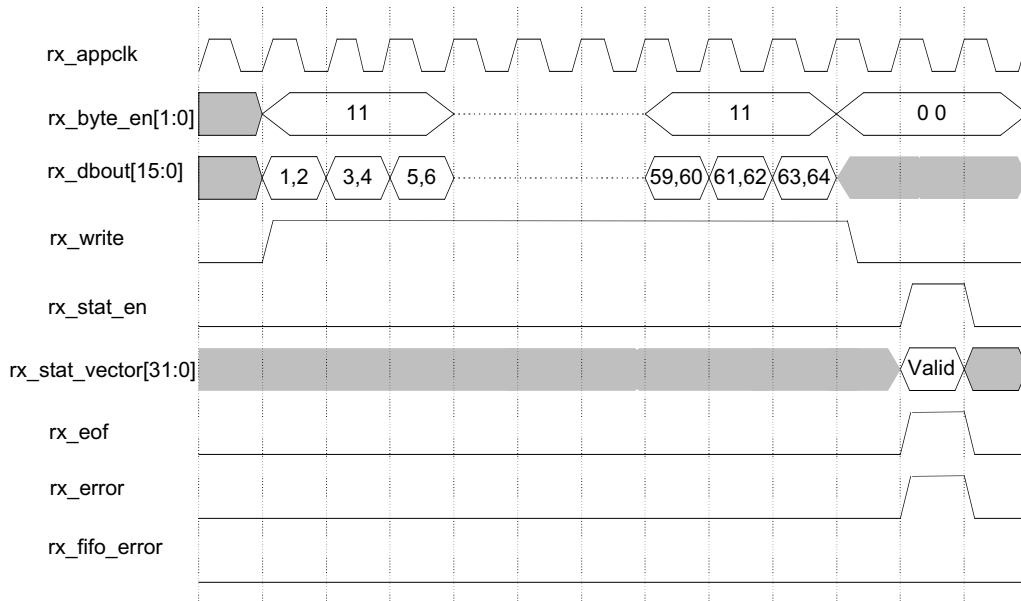
Figure 3. Reception of a 64-byte Frame Without Error



2. Reception of a 64-byte Frame with Error(s) - Rx MAC Application Interface (Gigabit Mode)

The signal `rx_error` is asserted to indicate that the 64-byte frame was received with error(s).

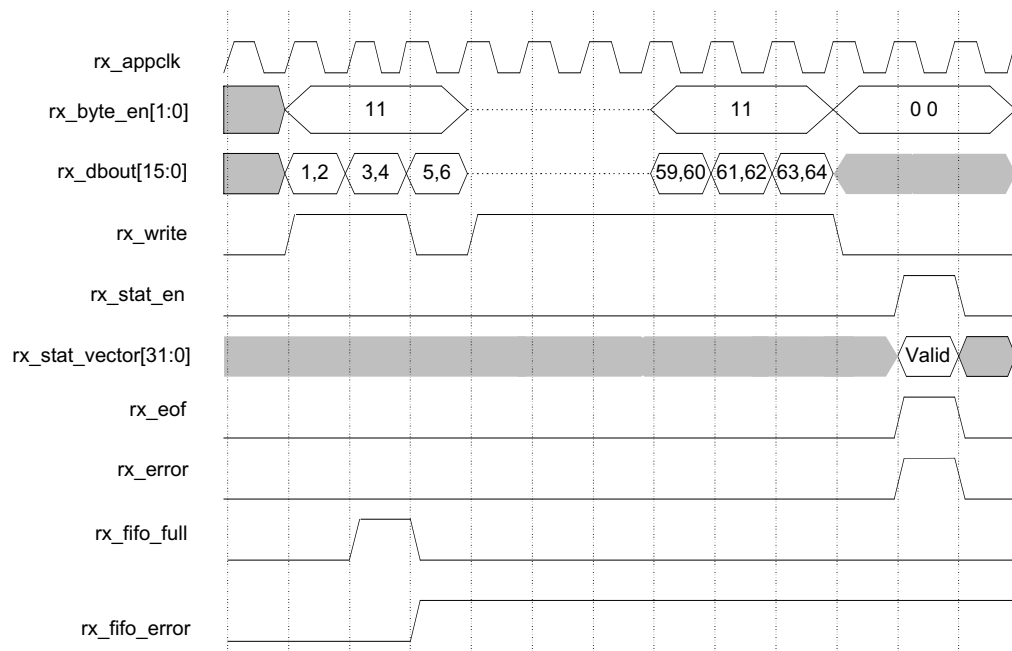
Figure 4. Reception of a 64-byte Frame with Error



3. Reception of a 64-Byte Frame with FIFO Overflow - Rx MAC Application Interface (Gigabit Mode)

The FIFO writing operation is suspended whenever an overflow condition occurs. When this condition occurs, the Tri-Speed MAC asserts `rx_fifo_error`. This signal should be sampled along with `rx_eof` in order to process the error condition.

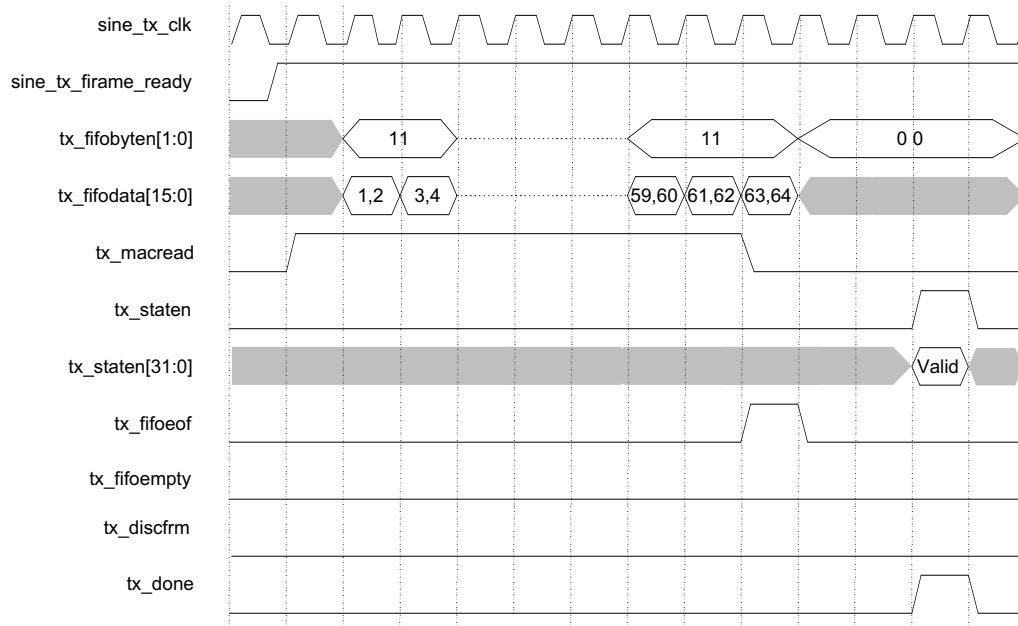
Figure 5. Reception of a 64-byte Frame with FIFO Overflow



4. Successful Transmission of a 64-Byte Frame -Tx MAC Application Interface (Gigabit Mode)

The assertion of `tx_fifoavail` indicates a frame is waiting to be transmitted. The Tri-Speed MAC reads the FIFO and the data is transmitted until `tx_fifoeof` is asserted. Once the frame is transmitted, `tx_staten` is asserted to qualify the statistic vector, `tx_statvec`. The signal `tx_done` is asserted to indicate a successful transmission. This is shown in Figure 6.

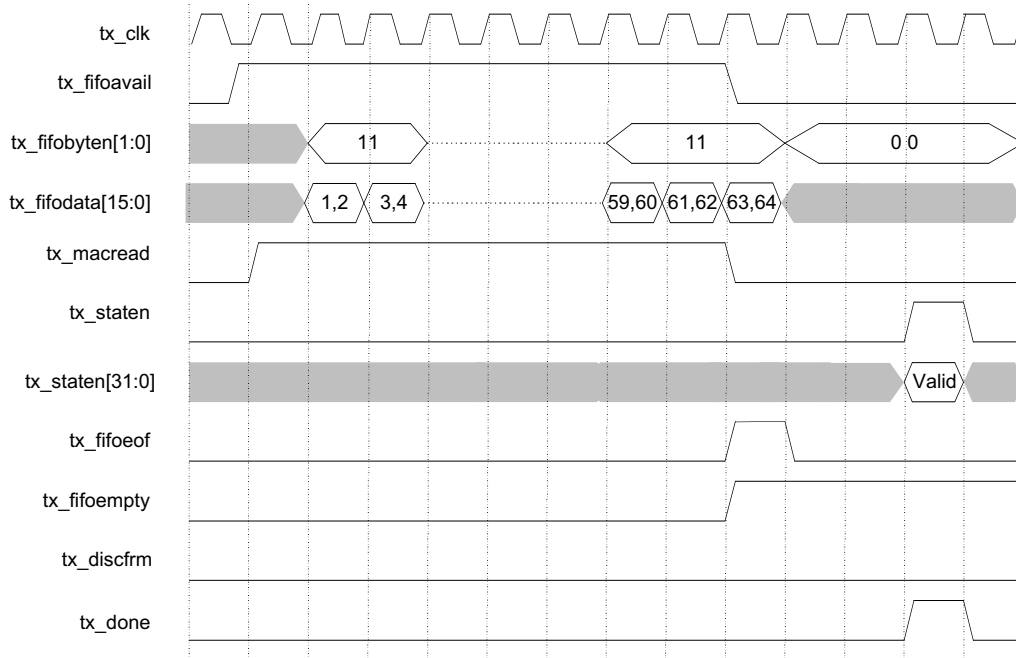
Figure 6. Transmission of a 64-Byte Frame without Error



5. Successful Transmission of a 64-byte Frame with FIFO Empty - Tx MAC Application Interface (Gigabit Mode)

tx_fifoempty is asserted along with tx_fifoef of to indicate that the complete 64-byte frame has been read. The frame is transmitted as a valid frame and tx_done is asserted at the end of transmission.

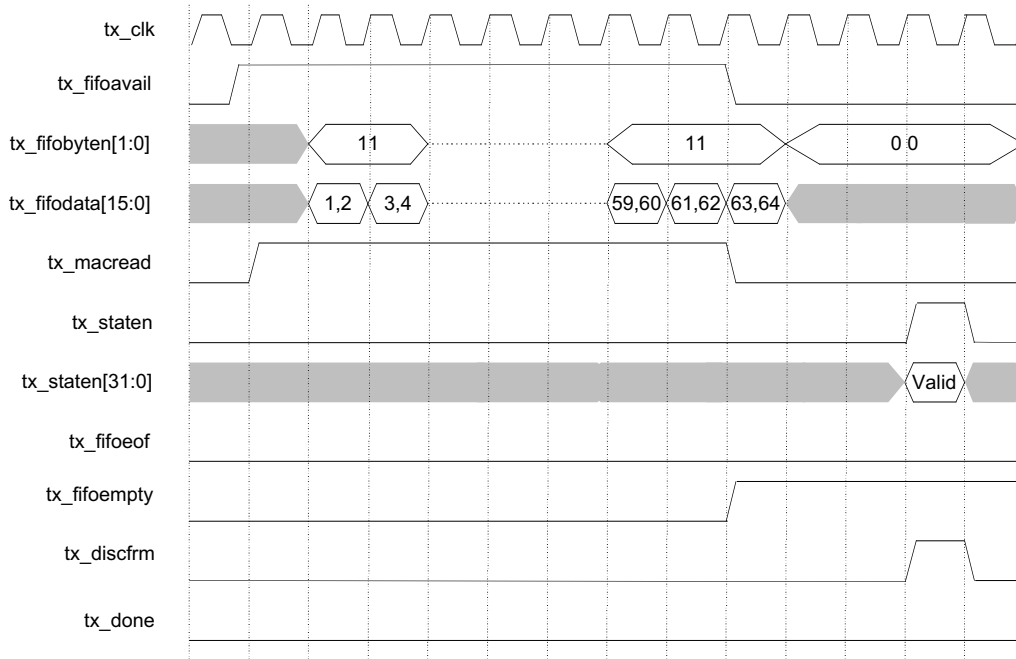
Figure 7. Successful Transmission of a 64-byte Frame with FIFO Empty



6. Aborted Transmission Due to FIFO Empty - Tx MAC Application Interface (Gigabit Mode)

If the `tx_fifoempty` is asserted while the Tx MAC is in the process of reading a frame, the MAC will stop reading the frame and assert `tx_disfrm` to indicate an erroneous transmission. The frame transmission is abandoned when this occurs.

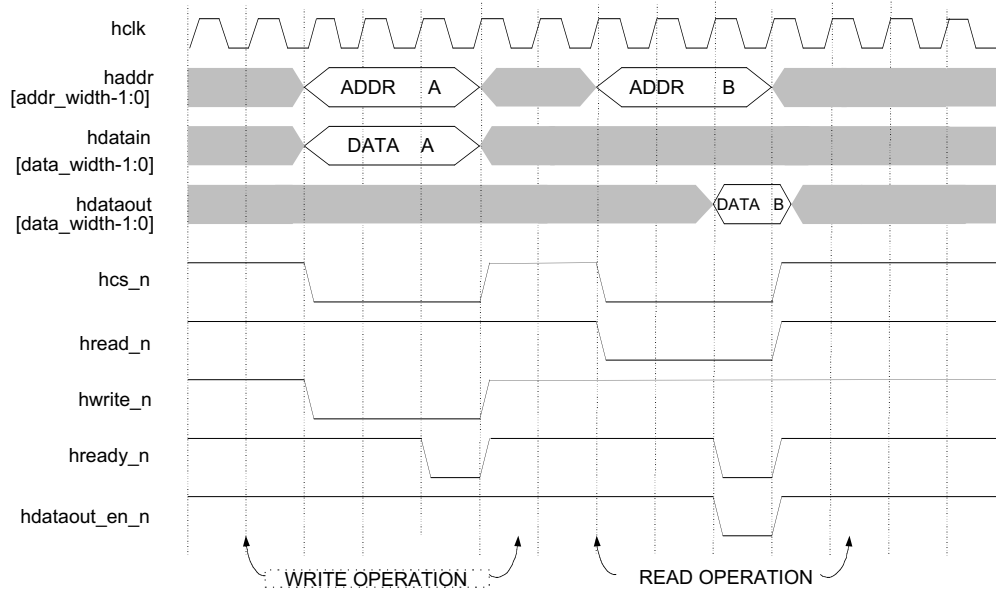
Figure 8. Aborted Transmission Due to FIFO Empty



7. Host Interface Read/Write Operation

During a write operation, haddr associated with hdatain, hcs_n and hwrite_n performs a write operation to an internal register. The end of transaction is indicated by assertion of hready_n. During a read operation, haddr associated with hcs_n and hread_n forms a write operation. The end of transaction is indicated by the assertion of hready_n and hdataout_en_n along with the valid read data on hdataout.

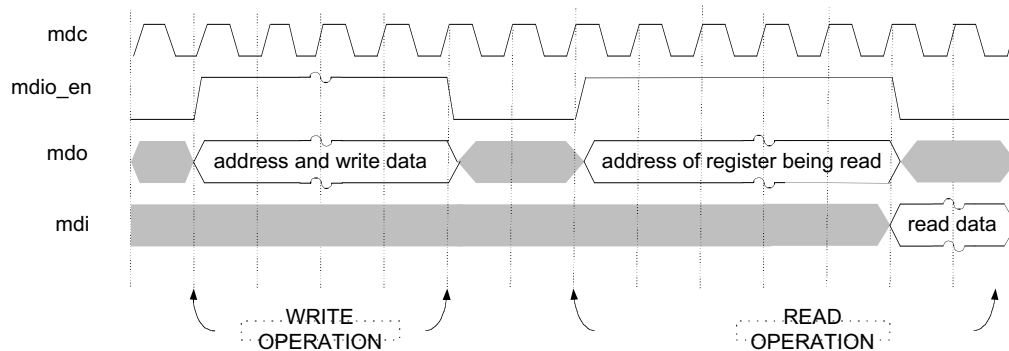
Figure 9. Host Interface read/write operation



8. Management Interface Read/Write Operation

During a write operation, mdio_en is asserted and the data is transmitted on mdo. During a read operation, mdio_en is asserted while the address is being transferred. Once this is done, it is de-asserted for rest of the transfer enabling the PHY to deliver data on mdi.

Figure 10. Management Interface Read and Write Operations

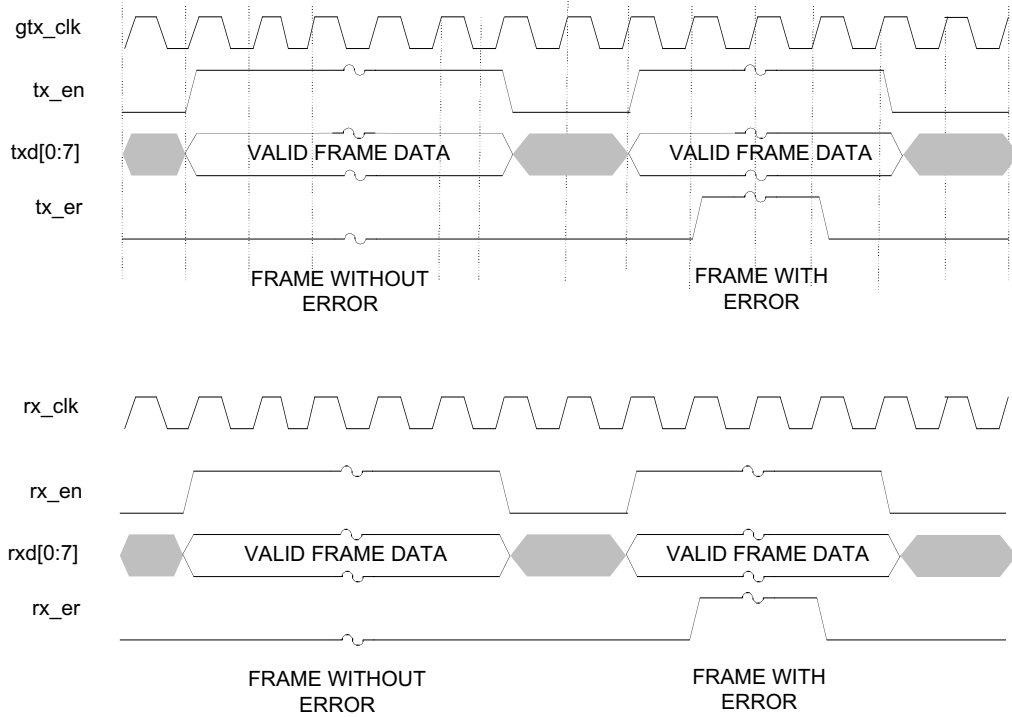


9. GMII Transmit and Receive Operations (Gigabit Mode)

txd and tx_en are driven synchronous to the gtx_clk during transmit operations. When the frame being transmitted has an error, tx_er is asserted.

When receiving data, rxd and rx_en are sampled on the rising edge of rx_clk. An error in the frame is indicated when rx_er is asserted.

Figure 11. GMII Transmit and Receive Operations



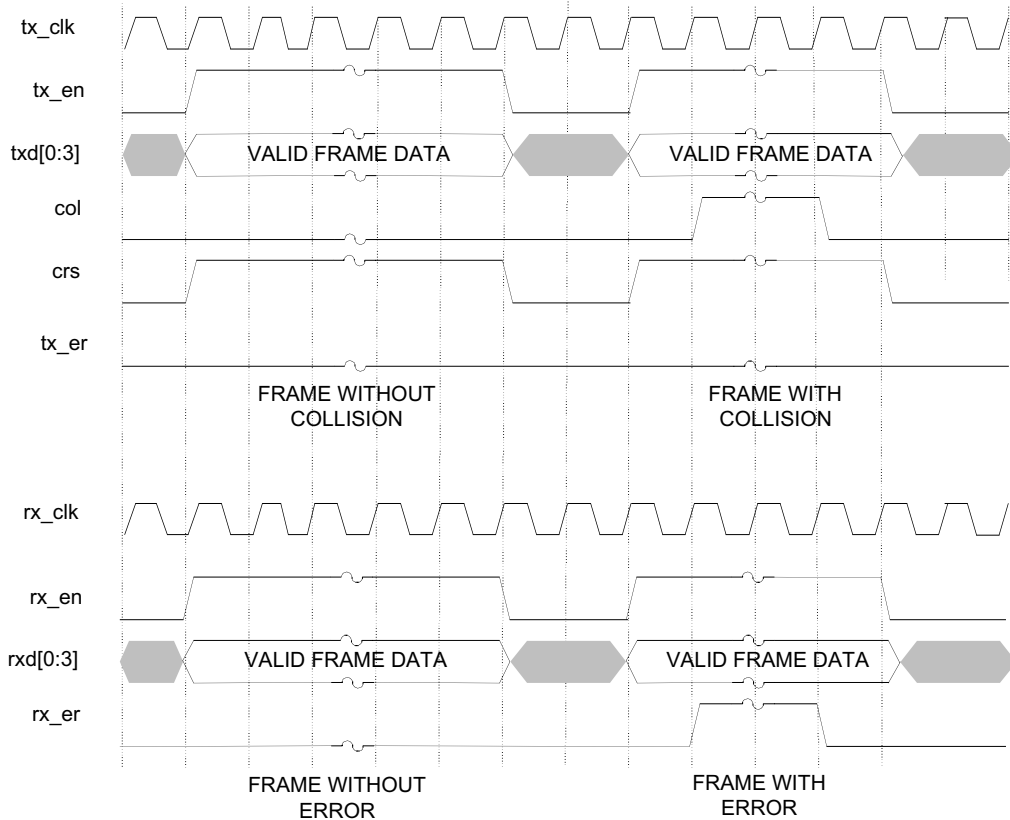
10. MII Transmit and Receive Operations (10/100 Mode)

On the transmit side, `txd` and `tx_en` are driven synchronous to the `gtx_clk`. `tx_er` is asserted to indicate that the frame being transmitted has an error.

On the receive interface, `rx_d` and `rx_en` are sampled on the rising edge of `rx_clk`. An error in the frame is indicated when `rx_er` is high when sampled on the rising clock edge.

`col` and `crs` are asynchronous signals, useful in the half-duplex mode only.

Figure 12. MII Transmit and Receive Operations



Custom Core Configurations

To request Tri-Speed MAC core configurations not available in the Evaluation Package, please contact your Lattice sales office.

Reference Information

The following documents provide more information on implementing this core:

- *ispLEVER® Software User Manual*
- *ispLeverCORE™ IP Module Evaluation Tutorial* available on the Lattice web site at www.latticesemi.com

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
 +1-503-268-8001 (Outside North America)
 e-mail: techsupport@latticesemi.com
 Internet: www.latticesemi.com

Appendix for ORCA® Series 4 FPGAs or FPSCs

Table 5. Performance and Resource Utilization¹

| Name of Parameter File | Mode (Mbps) | CPU Data Width | MIIM Module | LUTs | ORCA4 PFUs | Registers | I/Os | System EBR (RAM512) | f _{MAX} |
|-------------------------|-------------|----------------|-------------|------|------------|-----------|------|---------------------|--|
| ether_1gig_o4_3_001.lpc | 1000 | 16 | No | 1747 | 364 | 1313 | 201 | 1 | 125 MHz (GMII) |
| ether_fast_o4_3_002.lpc | 10/100 | 16 | Yes | 2581 | 548 | 1850 | 198 | 2 | 62.5 MHz sys_clk, 25MHz MII host clks (PHY side) |

1. Performance and utilization characteristics using ispLEVER software and targeting the ORCA4E04-2BA352C. When using this IP core in a different density, package, speed, or grade within the ORCA 4 family, performance may vary.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all 1000Mbps configurations of this core on ORCA Series 4 is ETHER-1GIG-04-N3. The Ordering Part Number (OPN) for all 10/100Mbps configurations of this core on ORCA Series 4 is ETHER-FAST-04-N3. Table 5 lists the netlist configurations that are available in the Evaluation Package for this core, which can be downloaded from the Lattice web site at www.latticesemi.com.

You can use the IPexpress™ software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

Appendix for ispXPGA® FPGAs

Table 6. Performance and Resource Utilization¹

| Name of Parameter File | Mode (Mbps) | CPU_Data_Width | MIIM Module | LUT4s | PFUs | Registers | I/Os | System EBRs | f _{MAX} |
|-------------------------|-------------|----------------|-------------|-------|------|-----------|------|-------------|--|
| ether_1gig_xp_1_001.lpc | 1000 | 16 | No | 2084 | 744 | 1502 | 201 | 2 | 125 MHz (GMII) |
| ether_fast_xp_1_002.lpc | 10/100 | 16 | Yes | 3545 | 1156 | 2179 | 198 | 4 | 62.5 MHz sys_clk, 25MHz MII host clks (PHY side) |

1. Performance and utilization characteristics using ispLEVER software and targeting the LFX500B-04FH516C device. The evaluation version of this IP core only works on this specific device density, package, and speed grade.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all 1000Mbps configurations of this core on ispXPGA is ETHER-1GIG-XP-N3. The Ordering Part Number (OPN) for all 10/100Mbps configurations of this core on ORCA ispXPGA is ETHER-FAST-XP-N3. Table 6 lists the netlist configurations that are available in the Evaluation Package for this core, which can be downloaded from the Lattice web site at www.latticesemi.com.

You can use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

Appendix for LatticeECP™ and LatticeEC™ FPGAs

Table 7. Performance and Resource Utilization¹

| Name of Parameter File | Mode (Mbps) | CPU_Data_Width | MIIM Module | LUT4s | SLICES | Registers | I/Os | System EBRs | f _{MAX} |
|-------------------------|-------------|----------------|-------------|-------|--------|-----------|------|-------------|--|
| ether_1gig_e2_3_001.lpc | 1000 | 16 | No | 1681 | 1318 | 1339 | 201 | 2 | 125 MHz (GMII) |
| ether_fast_e2_3_006.lpc | 10/100 | 16 | Yes | 2712 | 1892 | 1792 | 198 | 4 | 62.5 MHz sys_clk, 25MHz MII host clks (PHY side) |

1. Performance and utilization characteristics are generated using LFEC20E-4F672C in Lattice ispLEVER 4.1 software. When using this IP core in a different density, package, or speed grade, performance may vary.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all 1000Mbps configurations of this core on LatticeECP/EC is ETHER-1GIG-E2-N3. The Ordering Part Number (OPN) for all 10/100Mbps configurations of this core on LatticeECP/EC is ETHER-FAST-E2-N3. Table 7 lists the netlist configurations that are available in the Evaluation Package for this core, which can be downloaded from the Lattice web site at www.latticesemi.com.

You can use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

Appendix for LatticeXP™ FPGAs

Table 8. Performance and Resource Utilization¹

| Name of Parameter File | Mode (Mbps) | CPU_Data_Width | MIIM Module | LUT4s | SLICES | Registers | I/Os | System EBRs | f _{MAX} |
|-------------------------|-------------|----------------|-------------|-------|--------|-----------|------|-------------|--|
| ether_1gig_xm_3_001.lpc | 1000 | 16 | No | 1730 | 1328 | 1340 | 201 | 2 | 125 MHz (GMII) |
| ether_fast_xm_3_006.lpc | 10/100 | 16 | Yes | 3008 | 1892 | 1839 | 198 | 4 | 62.5 MHz sys_clk, 25MHz MII host clks (PHY side) |

1. Performance and utilization characteristics are generated using LFXP10C-4F388C in Lattice ispLEVER 5.0 software. When using this IP core in a different density, package, or speed grade, performance may vary.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all 1000Mbps configurations of this core on LatticeXP is ETHER-1GIG-XM-N3. The Ordering Part Number (OPN) for all 10/100Mbps configurations of this core on LatticeXP is ETHER-FAST-XM-N3. Table 8 lists the netlist configurations that are available in the Evaluation Package for this core, which can be downloaded from the Lattice web site at www.latticesemi.com.

You can use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.

Appendix for LatticeSC™ FPGAs

Table 9. Performance and Resource Utilization¹

| Name of Parameter File | Mode | CPU_Data_Width | MIIM Module | LUT4s | SLICES | Registers | I/Os | System EBRs | f _{MAX} (MHz) |
|-------------------------|--------|----------------|-------------|-------|--------|-----------|------|-------------|--|
| ether_1gig_sc_3_001.lpc | 1000 | 16 | No | 1929 | 1301 | 1333 | 201 | 1 | 125 MHz (GMII) |
| ether_fast_sc_3_006.lpc | 10/100 | 16 | Yes | 2856 | 1910 | 1831 | 198 | 2 | 62.5 MHz sys_clk, 25MHz MII host clks (PHY side) |

1. Performance and utilization characteristics are generated using LFSC3GA25E-5F900C in Lattice ispLEVER 5.1 SP2 software. When using this IP core in a different density, package, or speed grade, performance may vary.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all 1000Mbps configurations of this core on LatticeSC is ETHER-1GIG-SC-N3. The Ordering Part Number (OPN) for all 10/100Mbps configurations of this core on LatticeSC is ETHER-FAST-SC-N3. Table 9 lists the netlist configurations that are available in the Evaluation Package for this core, which can be downloaded from the Lattice web site at www.latticesemi.com.

You can use the IPexpress software tool to help generate new configurations of this IP core. IPexpress is the Lattice IP configuration utility, and is included as a standard feature of the ispLEVER design tools. Details regarding the usage of IPexpress can be found in the IPexpress and ispLEVER help system. For more information on the ispLEVER design tools, visit the Lattice web site at: www.latticesemi.com/software.