LatticeMico32 Tri-Speed Ethernet Media Access Controller

The LatticeMico32 Tri-Speed Ethernet media access controller (MAC) peripheral is an open-source IP core that includes all logic blocks necessary to enable the LatticeMico32 to interface seamlessly to an Ethernet PHY. The peripheral core includes an ispLEVER-generated Tri-Speed MAC IP core, parameterized 2, 4, or 8-kilobyte receive and transmit FIFO buffers, 32-bit receive and transmit statistics counter registers, system control and status registers, and selectable single or dual OpenCores 32-bit WISHBONE-compatible system-on-a-chip (SoC) slave interconnects used to interface to these blocks. The LatticeMico32 Tri-Speed MAC peripheral core, in conjunction with LatticeMico32 software (TCP/IP stack), provides the means for the LatticeMico32 software to interface easily to a fast or Gigabit Ethernet network.

Version

This document describes version 3.3 of the LatticeMico32 Tri-Speed Ethernet MAC that supports the 3.3 version of the IP Express Tri-speed MAC IP core.

Features

The LatticeMico32 Tri-Speed MAC peripheral includes the following features:

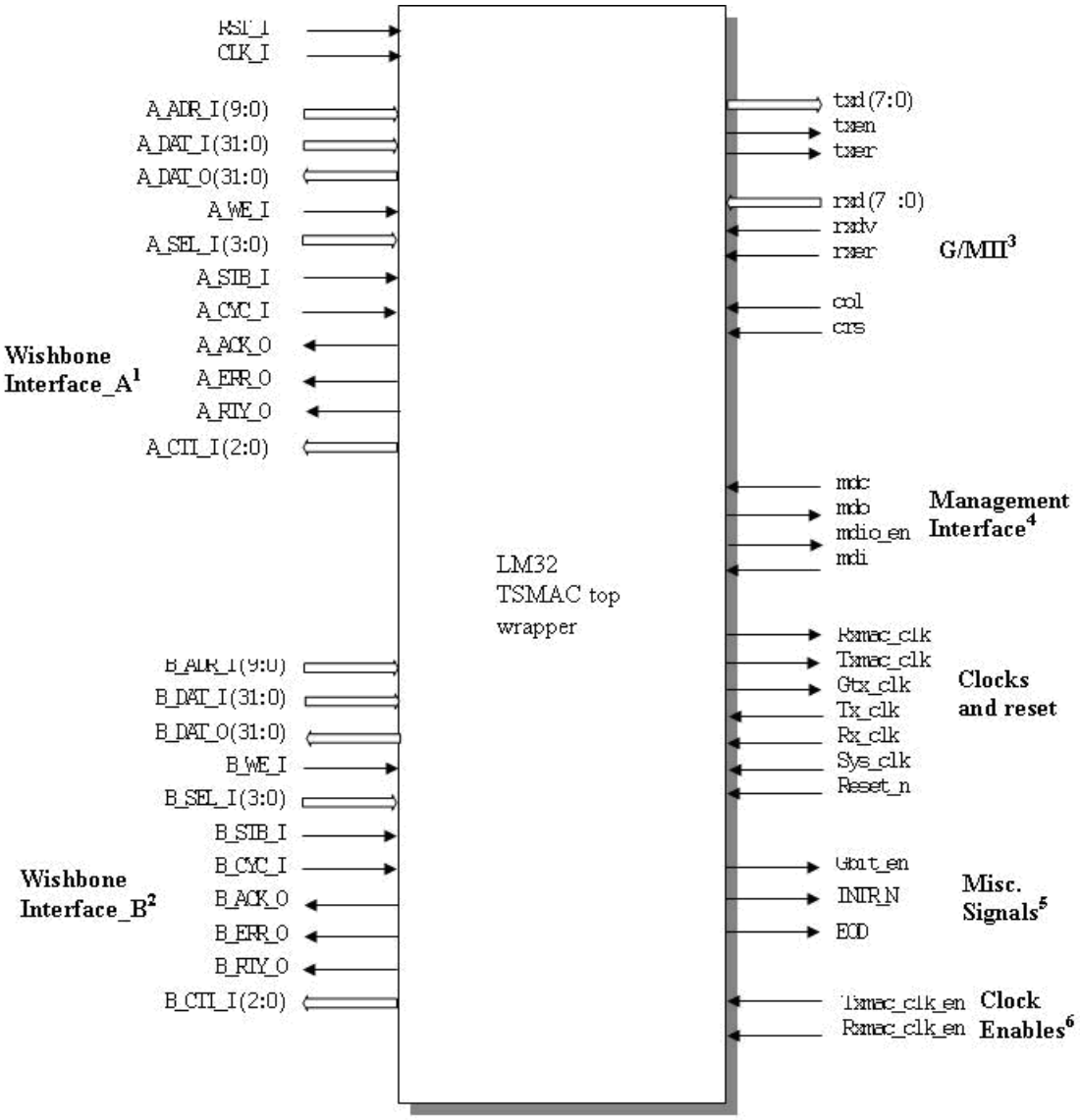
- Compliant to IEEE 802.3z standard
- Selectable single or dual 32-bit WISHBONE-compatible slave interfaces
- Single Wishbone Interface Mode – All registers are mapped to a single Wishbone Interface A., supporting Classic cycles.
- Dual Wishbone Interface Mode – Only Rx_data and Tx_data FIFOs are mapped to Wishbone Interface B, supporting Burst or Classic cycles. All

other registers are mapped to Wishbone Interface A., supporting Classic cycles.

- Optional Loopback module.
- Optional Management Interface (MDIO).
- Selectable MAC interfacing modes:
  - Classic Tri-Speed MAC with G/MII
  - Gbit MAC with GMII
  - SGMII Easy connect with “GMII”
- WISHBONE-accessible configurable (2, 4 or 8- kilobyte) transmit and receive FIFO buffers
- Optional WISHBONE-accessible 32-bit transmit and receive statistics counters
- WISHBONE-accessible internal Tri-Speed MAC core registers
- WISHBONE-accessible system control and status registers
- MAC Supports the following:
  - Multicast address filtering
  - Programmable inter-packet gap (IPG)
  - Full- and half-duplex operation in 10/100 mode
  - Full-duplex control using pause frames
  - VLAN tagged frames
  - Automatic retransmission on collision
  - Automatic padding of short frames
  - Multicast and broadcast frames
  - Optional FCS transmission and reception
  - Optional MII management interface module
  - Jumbo frames up to 8192 bytes

Figure 1 shows all the interfaces and signal ports in the LatticeMico32 Tri-Speed MAC.

Figure 1: LatticeMico32 Tri-Speed MAC



Notes on Figure 1

[1] Wishbone interface A is always present regardless of any options set.

[2] Wishbone interface B is present only if the dual Wishbone option is set.

[3] The col and crs signals are not present if the Gigabit option is set.

[4] These signals are only present if the MIIM option is set.

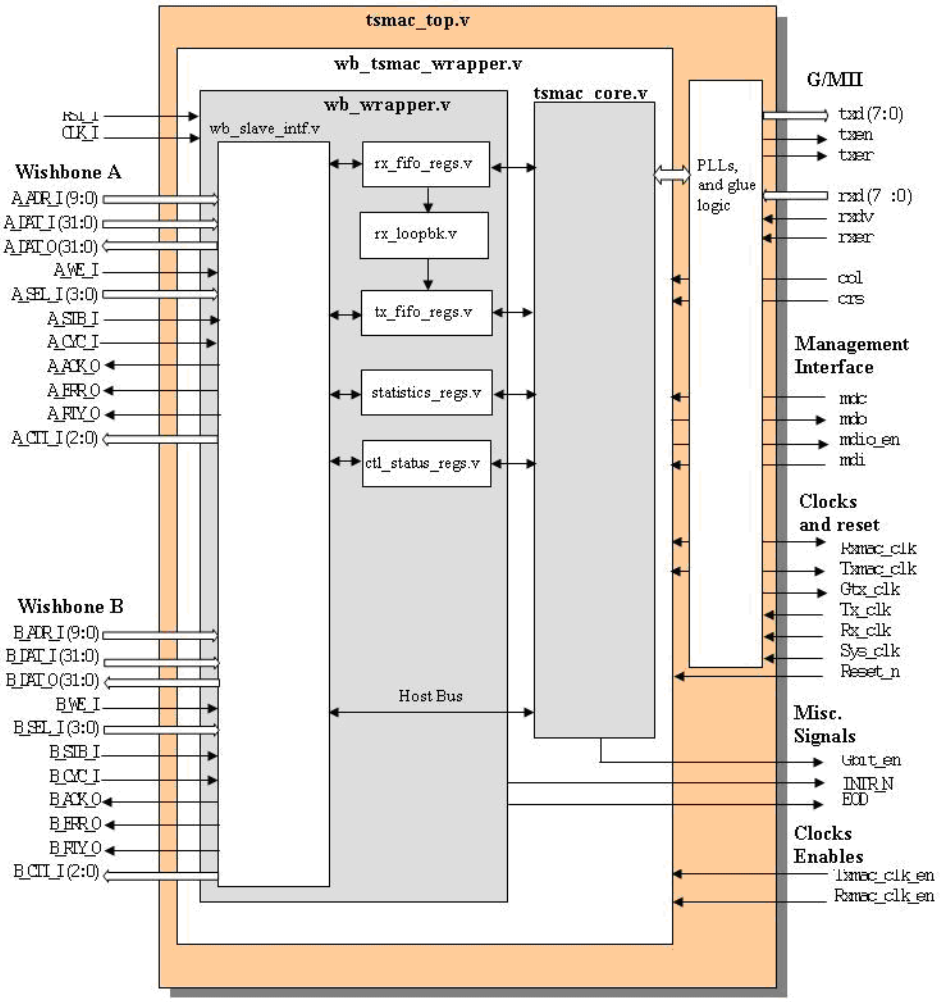
[5] EOD is only present if the EOD_TAG option is set.

[6] These signals are only present if the “SGMII Easy connect” option is set.

Functional Description

Figure 2 shows a block diagram of the tsmac top wrapper with its different levels of hierarchy. Figure 3 and Figure 4 show a more functional representation of the basic architecture of the wrapper design.

Figure 2: Block Diagram of the LatticeMico32 Tri-Speed MAC



Basic Architecture

The basic architecture of the LatticeMico32 Tri-Speed MAC with a single Wishbone interface is shown in Figure 3. The basic architecture of the LatticeMico32 Tri-Speed MAC with dual Wishbone interfaces and a Loopback module is shown in Figure 4.

- The blue dashed line shows the data flow to and from the G/MII and external memory.
- The orange dashed line shows the data flow to and from the LatticeMico32 microprocessor and the external memory. This flow is defined as the LatticeMico32 data path.
- The green dashed line shows the control and status data flow to and from the Tri-Speed MAC and the LatticeMico32 microprocessor. This flow is defined as the Tri-Speed MAC control path.
- The red dashed line shows the control and status data flow to and from the registers and the LatticeMico32 microprocessor. This flow is defined as the register control path.
- The Rx data path is defined as the Ethernet data from the G/MII to memory, and the Tx data path is defined as the Ethernet data from the memory to G/MII.
- The purple dashed line shows the Rx Ethernet data from the receive FIFOs Looped back to the Tx FIFOs.

Figure 3: Architecture of the LatticeMico32 Tri-Speed MAC with Single WB

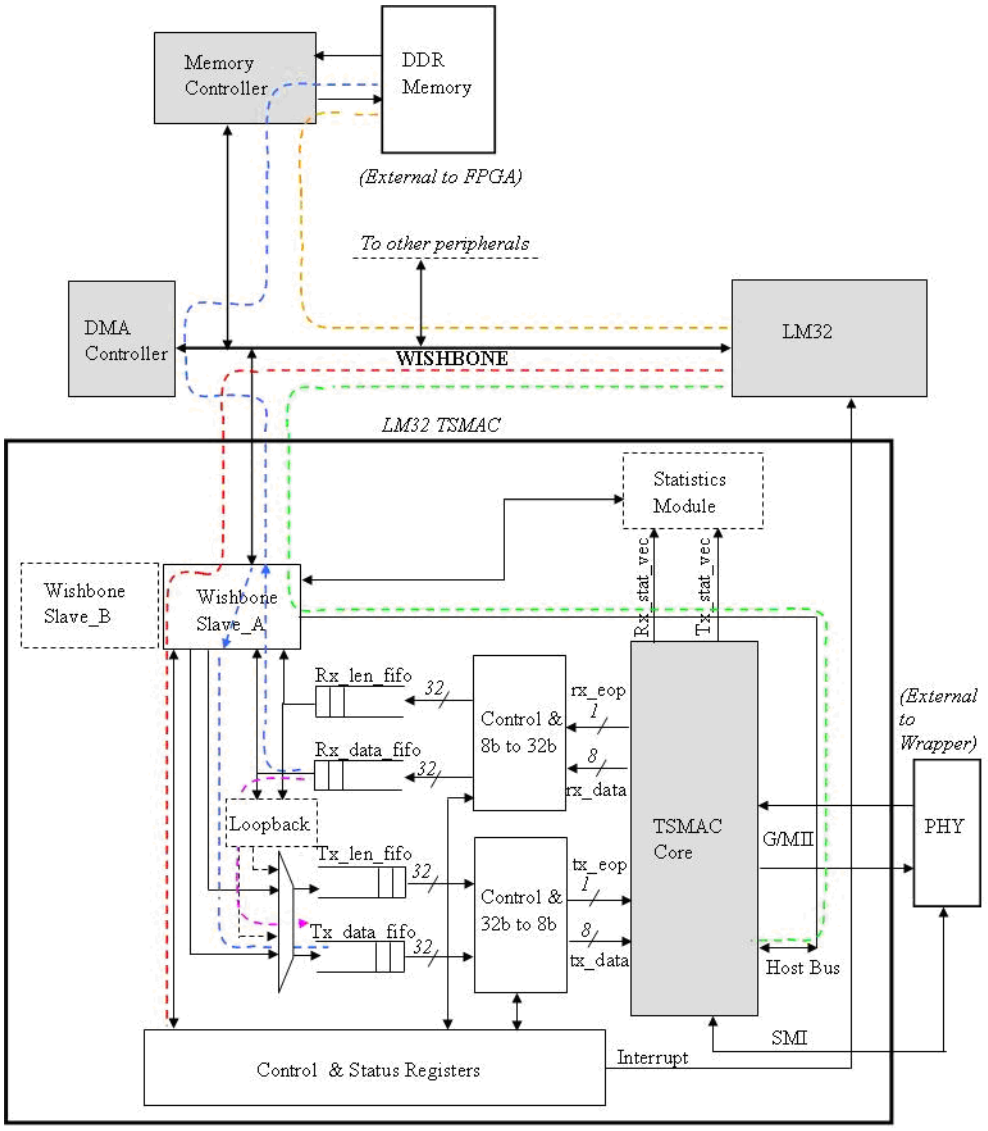
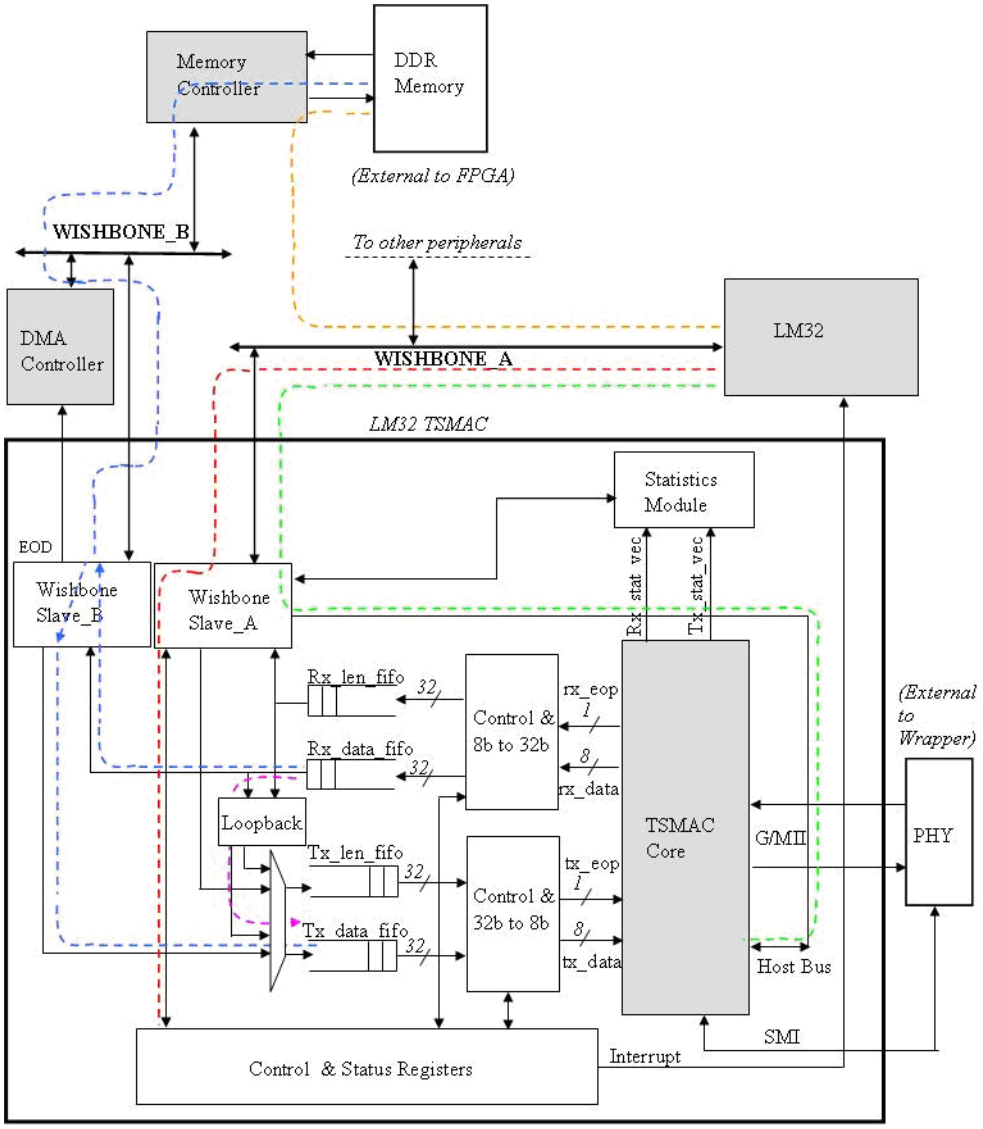


Figure 4: Architecture of the LatticeMico32 Tri-Speed MAC with dual WB



Basic Receive Flow

The basic receive (Rx) flow is as follows:

1. A receive packet is received and processed by the Tri-Speed MAC.
2. The Tri-Speed MAC checks to see if there is space available in the Rx_data FIFO. If there is, it begins writing the packet into the Rx_data FIFO. If the Rx_data FIFO is almost full, an interrupt is generated to the LatticeMico32 microprocessor, and the Tri-Speed MAC may assert the ignore_next_pkt bit in the SYS_CTL register (the receive Tri-Speed MAC drops packets when this signal is asserted), as well as signal the transmit Tri-Speed MAC to transmit pause frames to PHY by setting the tx_sndpausreq bit in the SYS _CTL register.

3. The Control-and-8b-to-32b block converts the 8-bit frame data to 32 bits and conditions the write enable to the 32-bit FIFO to write the data into the data FIFO appropriately. The length of the frame is calculated from the write enable and EOP.
4. After the frame is completely written into the Rx_data FIFO, the Control-and-8b-to-32b block writes the length of the frame into the Rx_len FIFO and sends an interrupt to the LatticeMico32 microprocessor, indicating that a receive frame is ready. A Rx_frame counter register is also incremented.
5. Once the LatticeMico32 microprocessor services the interrupt, it must read the Rx_frame counter register to make sure a receive frame is available (if interrupts are not desired, the LatticeMico32 microprocessor can just poll the Rx_frame counter to see if there are any available receive frames). The LatticeMico32 microprocessor then reads the Rx_len FIFO to determine how many bytes need to be read from the Rx_data FIFO. Once the length has been read, the LatticeMico32 microprocessor can set up a DMA transfer from the Rx_data FIFO to the DDR memory, or read the Rx_data FIFO and write to the memory itself. Once the Rx_len FIFO is read, the Rx_frame counter is decremented.

Note

When dual Wishbone interfaces are selected all registers, excluding the Rx_data FIFO and Tx_data FIFO, are mapped to the Wishbone A interface. Therefore, the Rx_len FIFO is accessible only via the Wishbone A Interface while the Rx_data FIFO is accessible only via the Wishbone B Interface.

When a single Wishbone interface is selected all registers are mapped to the Wishbone A interface. Therefore, both the Rx_len FIFO and the Rx_data FIFO are accessible via the Wishbone A Interface.

Wishbone interface A supports only classic cycles, while Wishbone interface B supports both classic and constant address burst cycles.

Basic Transmit Flow

The basic transmit (Tx) flow is as follows:

1. A transmit packet is ready to be transmitted by the LatticeMico32 microprocessor.
2. The LatticeMico32 microprocessor first checks to see if there is space available in the Tx_data FIFO. If there is, it begins writing the packet into the Tx_data FIFO (this may be a DMA transfer).

   Note

   The LatticeMico32 microprocessor checks the status of the Tx_data FIFO by reading the Tx_fifo_status register or the Tx_frame counter register. If the Tx_data FIFO is almost full or almost empty, an interrupt is generated to the LatticeMico32 microprocessor. In addition, the Tx_fifo_status register indicates the almost full or empty status.

3. After the frame is completely written into the Tx_data FIFO, the LatticeMico32 microprocessor writes the length of the frame into the Tx_len FIFO. When the length is written into the Tx_len FIFO, a Tx_frame

counter register is incremented. The Control-and-32b-to-8b block polls the Tx_frame counter register to see if a frame is available to be transmitted.

4. If a frame is available to be transmitted, the Control-and-32b-to-8b block reads the length of the frame from the Tx_len FIFO and indicates to the Tri-Speed MAC that the Tx FIFO is not empty by de-asserting the Tx_fifo empty signal and by asserting the tx_fifoavail signal.
5. When the Tri-Speed MAC asserts the tx_macread signal, the Control-and-32b-to-8b block converts the 32-bit frame data to 8 bits and conditions the read enable to the 32-bit FIFO to read the data out of the FIFO appropriately. At the end of the frame, the EOP is asserted in the correct location by using the frame length information from the Tx_len FIFO. The Tx_frame counter register is also decremented, and an interrupt is generated to the LatticeMico32 microprocessor to indicate that the frame was transmitted.

Note

When dual Wishbone interfaces are selected all registers, excluding the Rx_data FIFO and Tx_data FIFO, are mapped to the Wishbone A interface. Therefore, the Tx_len FIFO is accessible only via the Wishbone A Interface while the Tx_data FIFO is accessible only via the Wishbone B Interface.

When a single Wishbone interface is selected all registers are mapped to the Wishbone A interface. Therefore, both the Tx_len FIFO and the Tx_data FIFO are accessible via the Wishbone A Interface.

Wishbone interface A supports only classic cycles, while Wishbone interface B supports both classic and constant address burst cycles.

Loopback Flow

1. Packets are received and processed by the MAC as described in steps 1-4 of the Basic Receive Flow.
2. If the Loopback module is present and the Loopback control register bit is enabled, packets are extracted autonomously by the Loopback module (the loopback module first reads the Rx_len FIFO and then the Rx_data FIFO).
3. The Loopback module then writes the packets into the Tx_data FIFO in a similar manner as steps 3-5 of the Basic Transmit Flow. Note the Tx_len FIFO is written autonomously by the Loopback module.

Note

When in Loop back mode reading and writing to the Rx /Tx Len and data FIFOs via the Wishbone interface is disabled, since the Loop back module controls access to these FIFOs.

The Loop back mode may be useful as a debug mode to test the integrity of the MAC interface. Specifically if the MAC is configured correctly and can receive Ethernet packets and transmit looped back packets.

Configuration

The following sections describe the graphical user interface (UI) parameters, the hardware description language (HDL) parameters, and the I/O ports that you can use to configure and operate the LatticeMico32 Tri-Speed MAC.

UI Parameters

Table 1 shows the UI parameters available for configuring the LatticeMico32 Tri-Speed MAC through the Mico System Builder (MSB) interface.

Table 1: LatticeMico32 Tri-Speed MAC UI Parameters

Dialog Box Option	Description	Allowable Values	Default Value
Instance Name	Specifies the name of the Tri-Speed MAC instance.	Alphanumeric and underscores	ts_mac_core
Base Address	Specifies the base address for the Tri-Speed MAC instance. The minimum boundary alignment is 0X80.	0X80000000 – 0XFFFFFF80 If other components are included in the platform, the range of allowable values will vary.	0X80000000
Size	Specifies the depth size of the Tri-Speed MAC FIFOs, in bytes.	8192, 4096, 2048	8192
TxRx_FIFO_Depth	Specifies the width of the Tx and Rx FIFOs. Note that the width of these FIFOs is always 32 bits.	32	32
Number of Wishbone interfaces	Specifies the number of wishbone slave interfaces the LatticeMico32 Tri-Speed MAC will have.	1, 2	1
Include Loopback	Determines whether a parameter is set to include the rx_loopback.v module in the wb_wrapper.v module.	1, 0	0
Enable EOD	Determines whether a parameter is set that enables the EOD signal to be punched thru the wrapper as a primary output.	1, 0	0
MAC Operation Mode	Specifies which Mode the Tri-Speed MAC instance will operate in.	Classic_TSMAC, Gbit_MAC, SGMII_easy connect	Classic_TSMAC
Include Statistics Regs	Determines whether a parameter is set to include the statistics_regs.v module in the wb_wrapper.v module. These statistics counter registers are described in the *IPUG51 - Tri-speed Ethernet Media Access Controller User's Guide* in the sections on the transmit and receive statistics vectors.	1, 0	1

Table 1: LatticeMico32 Tri-Speed MAC UI Parameters (Continued)

Dialog Box Option	Description	Allowable Values	Default Value
Use MDIO Interface	Includes the MDIO interface in the Tri-Speed MAC IP core generated by IPexpress. The MDIO interface is a standard IEEE 802.3 serial management interface consisting of an mdc signal (low-speed clock), an mdi (data_in) signal, an mdo (data_out) signal, and an mdio_en (bidirectional enable) signal. This interface allows the Tri-Speed MAC IP core to interface to other Ethernet devices, like a PHY device, and configure and monitor its registers. Access to the MDIO interface on the TSMAC IP core client side is through the host bus interface described in the *IPUG51 - Tri-speed Ethernet Media Access Controller User's Guide*. When the TSMAC wrapper is used with the WISHBONE interface, access to the MDIO interface is through the WISHBONE bus. You should select this option if "use MII" is enabled in IPexpress. You should not select this option if "use MII" is disabled in IPexpress.	1, 0	1
IPexpress Interface			
ispLEVER Project	Specifies the path and name of the ispLEVER project to which to add the Tri-Speed MAC controller instance.		
Generated NGO File	Specifies the path and name of the .ngo file containing the Tri-Speed MAC controller information. If you do not want to specify the location of the .ngo file in this box, LatticeMico32 System will place this information in the box after you click the Launch IPexpress button.		
Launch IPexpress	Starts IPexpress and generates the Tri-Speed MAC controller.		

HDL Parameters

Table 2 lists the parameters that appear in the HDL.

Table 2: LatticeMico32 Tri-Speed MAC HDL Parameters

Parameter Name	Description	Allowable Values
STAT_REGS	Sets a parameter to include the statistics_regs.v module in the wb_wrapper.v module. These statistics counter registers are described in the *IPUG51 - Tri-speed Ethernet Media Access Controller User's Guide* in the sections on the transmit and receive statistics vectors.	1, 0
MULT_WB	Sets the number of Wishbone slave interfaces (single or dual).	1, 0
EOD_TAG	Enables the EOD signal to be punched thru to the top.	1, 0
LOOPBACK	Sets a parameter to include the rx_loopback.v module in the wb_wrapper.v module.	1, 0
MIIM_MODULE	Includes the MDIO interface in the Tri-Speed MAC IP core generated by IPexpress. The MDIO interface is a standard IEEE 802.3 serial management interface consisting of an mdc signal (low-speed clock), an mdi (data_in) signal, an mdo (data_out) signal, and an mdio_en (bidirectional enable) signal. This interface allows the Tri-Speed MAC IP core to interface to other Ethernet devices, like a PHY device, and configure and monitor its registers. Access to the MDIO interface on the TSMAC IP core client side is through the host bus interface described in the *IPUG51 - Tri-speed Ethernet Media Access Controller User's Guide*. When the TSMAC wrapper is used with the WISHBONE interface, access to the MDIO interface is through the WISHBONE bus.	1, 0

I/O Ports

Table 3 describes the input and output ports of the LatticeMico32 Tri-Speed MAC.

Table 3: LatticeMico32 Tri-Speed MAC Signals

Signal	Type	Active State	Description
sys_clk	Input	N/A	System Clock. 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 125 MHz. Note in the Gigabit mode the Tri-speed MAC's txmac_clk is derived from this clock.
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 while in the 10/100 mode, the corresponding rx_clk frequency is 2.5/25 MHz respectively. In the 10/100 mode rx_clk is divided by two to provide the clock (rxmac_clk) to the Receive MAC section. In the 1G mode this clock is provided directly to the Receive MAC section. The receive signals at the GMII interface are always synchronous to rx_clk.
Tx_clk	Input	N/A	Transmit Clock. This clock is used in the 10/100 Mbps mode only. The Tx MAC, Tx MAC application interface and the MII are synchronous to this signal. This clock has a frequency of 2.5/25 MHz for 10/100 Mbps operation respectively. Note in the 10/100 mode tx_clk is divided by two to provide the clock (txmac_clk) to the transmit MAC section. In the 10/100 mode the transmit signals at the GMII interface are always synchronous to tx_clk.
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. This clock is derived from the sys_clk.
rxmac_clk	Output	N/A	The receive Tri-Speed MAC application interface clock is used by the Tri-Speed MAC and internal logic. All outputs driven by the receive Tri-Speed MAC to the internal interfaces (receive FIFO to MAC and so forth) are synchronous to this clock. This clock's frequency is 125,12.5, or 1.25 MHz, depending on whether the mode is 1G, 100, or 10, respectively. **Note**: This clock can be viewed as a byte clock, because all receive Tri-Speed MAC bytes are aligned with this clock. This clock is derived from the system G/MII Rx_clk.

Table 3: LatticeMico32 Tri-Speed MAC Signals (Continued)

Signal	Type	Active State	Description
txmac_clk	Output	N/A	The transmit Tri-Speed MAC application interface clock is used by the Tri-Speed MAC and internal logic. All internal inputs to the transmit Tri-Speed MAC on the internal interface (transmit FIFO to MAC and so forth) should be synchronous to this clock. This clock's frequency is 125,12.5, or 1.25 MHz, depending on whether the mode is 1G, 100, or 10, respectively. **Note**: This clock can be viewed as a byte clock, because all transmit Tri-Speed MAC bytes should be aligned with this clock. This clock is derived from the system sys_clk or Tx_clk (1G or 10/100, respectively).
rxmac_clk_en	Input		Receive Clock Enable input signal is a clock enable used only in the "SGMII easy connect" mode. The SGMII_PCS IP core drives this signal. The clock enable is always high for 1G operation. For 100 Mbs operation the clock enable is asserted high once every ten (125Mhz) clocks, and for 10 Mbs operation the clock enable is asserted high once every hundred (125Mhz) clocks. **Note**: This signal is only present when the Tri-speed MAC is configured to operate in the "SGMII easy connect" mode.
txmac_clk_en	Input		Transmit Clock Enable input signal is a clock enable used only in the "SGMII easy connect" mode. The SGMII_PCS IP core drives this signal. The clock enable is always high for 1G operation. For 100 Mbs operation the clock enable is asserted high once every ten (125Mhz) clocks, and for 10 Mbs operation the clock enable is asserted high once every hundred (125Mhz) clocks. **Note**: This signal is only present when the Tri-speed MAC is configured to operate in the "SGMII easy connect" mode.
gbit_en	Output	High	The CPU interface gigabit -mode-enabled signal, when high, is an indication from the Tri-Speed MAC that the 1-gigabit mode is enabled. It reflects the state of bit 0 of the Tri-Speed MAC mode register.
mdc	Input	N/A	The management data clock is used only when the management interface module is implemented.
INTR_N	Output	Low	The interrupt to the LatticeMico32 microprocessor (see "Register Definitions" on page 18 and "Interrupt Hierarchy" on page 46).

Table 3: LatticeMico32 Tri-Speed MAC Signals (Continued)

Signal	Type	Active State	Description
EOD	Output	Low	End of data indicator (End of packet) used to facilitate bursting data out of the Rx_data FIFO. **Note**: This signal is only present if the EOD_TAG parameter is set.
reset_*n*	Input	Low	The reset is an active low signal that resets the internal registers and internal logic in the Tri-Speed MAC IP core. When activated, the I/O signals are driven to their inactive levels.
WISHBONE Interface			
RST_I	Input	High	The reset is an active high WISHBONE signal that resets the internal WISHBONE-related registers and internal logic. When activated, the I/O signals are driven to their inactive levels. You can invert this signal inside the wrapper and tie it to reset_*n*, if you wish.
CLK_I	Input	N/A	The WISHBONE clock input [CLK_I] coordinates all activities for the internal logic within the WISHBONE interconnect. All WISHBONE output signals are registered at the rising edge of [CLK_I]. All WISHBONE input signals are stable before the rising edge of [CLK_I.
A_ADR_I[11:0] B_ADR_I[11:0]	Input	N/A	The address input array [ADR_I()] is used to pass a binary address. The higher array boundary is specific to the address width of this wrapper, and the lower array boundary is determined by the data port size. In this wrapper, there is a 32-bit data port with byte granularity.
A_DAT_I[31:0] B_DAT_I[31:0]	Input	N/A	The data bus input array [DAT_I()] is used to pass binary data. The array boundaries are determined by the port size. The port size of this wrapper is 32 bits (for example, [DAT_I(31..0)]). See also the [DAT_O()] and [SEL_O()] signal descriptions.
A_DAT_O[31:0] B_DAT_O[31:0]	Output	N/A	The data output array [DAT_O()] is used to pass binary data. The array boundaries are determined by the port size. The port size of this wrapper is 32 bits (for example, [DAT_I(31..0)]). See also the [DAT_I()] and [SEL_O()] signal descriptions.
A_WE_I B_WE_I	Input	High	The write enable input. [WE_I] indicates whether the current local bus cycle is a read or write cycle. The signal is negated during read cycles and asserted during write cycles.

Table 3: LatticeMico32 Tri-Speed MAC Signals (Continued)

Signal	Type	Active State	Description
A_SEL_I(3:0) B_SEL_I(3:0)	Input	High	The select input array [SEL_I()] indicates where valid data is placed on the [DAT_I()] signal array during write cycles and where it should be present on the [DAT_O()] signal array during read cycles. The array boundaries are determined by the granularity of a port. In this wrapper, 8-bit granularity is used on a 32-bit port, so there is an array of four select signals with boundaries of [SEL_I(3..0)]. Each individual select signal correlates to one of four active bytes on the 32-bit data port. See also the [STB_I] signal descriptions.
A_STB_I B_STB_I	Input	High	The strobe input [STB_I], when asserted, indicates that the slave is selected. A slave responds to other WISHBONE signals only when this [STB_I] is asserted, except for the [RST_I] signal, which should always be responded to. The slave asserts either the [ACK_O], [ERR_O], or [RTY_O] signals in response to every assertion of the [STB_I] signal.
A_CYC_I B_CYC_I	Input	High	The cycle input [CYC_I], when asserted, indicates that a valid bus cycle is in progress. The signal is asserted for the duration of all bus cycles. For example, during a block transfer cycle, multiple data transfers can occur. The [CYC_I] signal is asserted during the first data transfer and remains asserted until the last data transfer.
A_ACK_O B_ACK_O	Output	High	The acknowledge output [ACK_O], when asserted, indicates the termination of a normal bus cycle. See also the [ERR_O] and [RTY_O] signal descriptions.
A_ERR_O B_ERR_O	Output	High	The error output [ERR_O] indicates an abnormal cycle termination. The source of the error and the response generated by the master is defined by the IP core supplier. See also the [ACK_O] and [RTY_O] signal descriptions.
A_RTY_O B_RTY_O	Output	High	The retry output [RTY_O] indicates that the interface is not ready to accept or send data and that the cycle should be retried. In this wrapper, the cycle should be retried after 20 CLK_I ticks. See also the [ERR_O] and [RTY_O] signal descriptions.

Table 3: LatticeMico32 Tri-Speed MAC Signals (Continued)

Signal	Type	Active State	Description
A_CTI_I[2:0] B_CTI_I[2:0]	Input	N/A	The cycle type identifier [CTI_IO()] address tag provides additional information about the current cycle. The master sends this information to the slave. The slave can use this information to prepare the response for the next cycle. CTI_O(2:0) description: ◆ 000 Classic cycle ◆ 001 Constant address burst cycle ◆ 010 Incrementing burst cycle ◆ 011 Reserved ◆ 100 Reserved ◆ 101 Reserved ◆ 110 Reserved ◆ 111 End-of-burst **Note**: Only the classic cycles are supported on Port_A, and both classic and constant address burst cycles (with End-of-burst) are supported on Port_B.
Management Interface Signals			
mdi	Input	High	The management data input is used to transfer information from the PHY to the management module.
mdo	Output	High	The management data output is used to transmit information from the management module to the PHY.
mdio_en	Output	High	The management data out enable is asserted whenever mdo is valid. It can be used to implement a bidirectional signal for mdi and mdo.
G/MII Signals			
txd[7:4] txd[3:0]	Output	High	txd[7:4] is the transmit data sent to the PHY chip – high nibble. In gigabit mode, these bits are the GMII txd[7:4] bits. They are clocked at 125 MHz (txmac_clk). These bits are not used in 10/100 mode. See the *IPUG51 - Tri-speed Ethernet Media Access Controller User's Guide*. txd[3:0] are the transmit data sent to the PHY chip – low nibble. In both gigabit mode and 10/100 mode, the txd [3:0] bits are the G/MII txd[3:0] bits. They are clocked by txmac_clk. In 1-gigabit mode, the txmac_clk rate is 125 MHz, and in the 10/100 mode, the clock rate is 1.25 MHz and 12.5 MHz, respectively. See the *IPUG51 - Tri-speed Ethernet Media Access Controller User's Guide.*

Table 3: LatticeMico32 Tri-Speed MAC Signals (Continued)

Signal	Type	Active State	Description
txen	Output	High	The transmit enable is asserted by the Tri-Speed MAC to indicate that the txd bus contains a valid frame.
txer	Output	High	The transmit error is asserted when the Tri-Speed MAC generates a coding error on the byte currently being transferred.
rxdv	Input	High	The receive data valid indicates that the data on the rxd bus is valid. For more information, see *IPUG51 - Tri-speed Ethernet Media Access Controller User's Guide*.
rxd[7:4] rxd[3:0]	Input	N/A	Receive data bus. Data is driven by the PHY on these lines and is valid whenever rxdv is asserted. rxd[7:4] are not used in 10/100 mode, only the low nibble bits rxd[3:0] used in 10/100 mode. See the *IPUG51 - Tri-speed Ethernet Media Access Controller User's Guide*.
rxer	Input	High	The receive data error signal is asserted by the external PHY device when it detects an error during frame reception. See the *IPUG51 - Tri-speed Ethernet Media Access Controller User's Guide*.
col	Input	High	The collision is an active-high signal that indicates that a collision occurred during transmission. This signal is valid for half-duplex operations in fast Ethernet (10/100) mode only. Otherwise, it is ignored. **Note**: These signals absent for the Gbit_MAC mode.
crs	Input	High	The carrier sense, when at logic high, indicates that the network has activity. Otherwise, it indicates that the network is idle. This signal is valid for half-duplex operations in fast Ethernet (10/100) mode only. **Note**: These signals absent for the Gbit_MAC mode.

Register Definitions

The LatticeMico32 Tri-Speed MAC has registers in four distinct memory mapped spaces:

- Tx_data, Rx_data, Tx_len, and Rx_len FIFO register
- Control and status registers
- Tri-Speed MAC IP core registers (registers inside the MAC IP core)
- Statistics counter registers

The Tx_data, Rx_data, Tx_len, and Rx_len FIFO registers are 32-bit data and length FIFOs, with fixed addresses. The LatticeMico32 microprocessor firmware accesses these FIFOs through the transmit and receive data paths.

Note

When Dual Wishbone interfaces are selected, the Rx_data and Tx_data FIFOs are the only registers mapped to interface B; all other registers reside on interface A . When a single Wishbone interface is selected, all registers reside on interface A

The control and status registers are 32-bit registers used to control the reception and transmission of Ethernet frames from and to the DDR memory and to and from the FIFOs. The LatticeMico32 microprocessor firmware accesses these control and status registers through the register control path.

The Tri-Speed MAC registers enable, configure, and monitor the Tri-Speed MAC. Since each Tri-Speed MAC register consists of two 8-bit registers (byte_0 and byte_1) on word boundaries, a single read or write to these registers through the 32-bit WISHBONE (Tri-Speed MAC control path) is an atomic operation that consists of two 8-bit register accesses, handled by a separate finite state machine. Two registers are used to access the Tri-Speed MAC registers:

- The MAC_REG_ADDR_RW register contains the read or write address of the register to be accessed (the address written to this register is on word boundaries), and a read/write control bit that indicates a read (low) or write (high) operation.
- The MAC_REG_DATA register contains the data read back or the data to write to the address specified in the MAC_REG_ADDR register.

A Tri-Speed MAC register read consists of the following:

- Writing the MAC_REG_ADDR_RW register with the appropriate address and with the read/write bit set to 0. Writing the MAC_REG_ADDR_RW register kicks off the atomic operation, and if an atomic operation is already in progress, the WISHBONE RTY_O (retry) signal is asserted to indicate that the Tri-Speed MAC host bus is busy and that the access will not be performed.
- Reading the MAC_REG_DATA register (one word read back). If the data from the atomic read is not yet valid, the WISHBONE RTY_O (retry) signal is asserted to indicate that the Tri-Speed MAC host bus is busy and that the access will not be performed.

A Tri-Speed MAC register write consists of: the following:

- Writing the MAC_REG_DATA register with the two bytes to be written. If an atomic operation is already in progress, the WISHBONE RTY_O (retry) signal is asserted to indicate that the Tri-Speed MAC host bus is busy and that the data just written will not be used.
- Writing the MAC_REG_ADDR_RW register with the appropriate address and with the read/write bit set to 1. Writing the MAC_REG_ADDR_RW register kicks off the atomic operation, and if an atomic operation is already in progress, the WISHBONE RTY_O (retry) signal is asserted to

indicate that the Tri-Speed MAC host bus is busy and that the access will not be performed.

Note

The additional processor clock cycles involved in writing to the Tri-Speed MAC registers should not be an issue since these registers are primarily intended to be configured once and then ignored.

The statistics counter registers are 32-bit registers used to keep received and transmitted packet statistics. The LatticeMico32 microprocessor firmware accesses these control and status registers through the register control path. The statistics counter registers reside in a separate module within the Tri-Speed MAC wrapper, and you can pull them out, if you wish, with the Include Statistics Regs option in the graphical user interface.

Data organization of the 32-bit port conforms to the little endian ordering shown in Table 4.

Table 4: 32-Bit Data Bus with 8-Bit (Byte) Granularity

	Address Range	Active Portion of Data Bus			
	ADR_I	DAT_I	DAT_I	DAT_I	DAT_I
	ADR_O	DAT_O	DAT_O	DAT_O	DAT_O
	(31..2)	(31..24)	(23..16)	(15..08)	(07..00)
	Active	SEL_I(3)	SEL_I(2)	SEL_I(1)	SEL_I(0)
	Select line	SEL_O(3)	SEL_O(2)	SEL_O(1)	SEL_O(0)
Byte Ordering	Little endian	Byte(3)	Byte(2)	Byte(1)	Byte(0)
		Byte(7)	Byte(6)	Byte(5)	Byte(4)

Table 5 describes the Tri-Speed MAC registers.

Table 5: LatticeMico32 Tri-Speed MAC Register Descriptions

Address [11:0] (**Note:** only [11:2] is decoded)	Register Name	Description	Size	Type
FIFO Registers				
0x000	RX_LEN _FIFO	Receive packet-length FIFO register	32 bits	Read only
0x004	RX_DATA_FIFO	Receive packet-data FIFO register See register description note.	32 bits	Read only
0x008	TX_LEN_FIFO	Transmit packet-length FIFO register	32 bits	Read/write

Table 5: LatticeMico32 Tri-Speed MAC Register Descriptions (Continued)

Address [11:0] (Note: only [11:2] is decoded)	Register Name	Description	Size	Type
0x00C	TX_DATA_FIFO	Transmit packet-data FIFO register See register description note.	32 bits	Read/write
0x010 – 0x0FC	Reserved			
Control and Status Registers				
0x100	VERID	Version ID register	32 bits	Read only
0x104	INTR_SRC	Interrupt source register	32 bits	Read only, clear on read
0x108	INTR_ENB	Interrupt enable register	32 bits	Read/write
0x10C	RX_STATUS	Receive status register	32 bits	Read only, clear on read
0x110	TX_STATUS	Transmit status register	32 bits	Read only, clear on read
0x114	RX_FRAMES_CNT	Receive frames counter register	32 bits	Read only
0x118	TX_FRAMES_CNT	Transmit frames counter register	32 bits	Read only
0x11C	RX_FIFO_TH	Receive FIFO threshold register	32 bits	Read/write
0x120	TX_FIFO_TH	Transmit FIFO threshold register	32 bits	Read/write
0x124	SYS_CTL	System control register	32 bits	Read/write
0x128	PAUSE_TMR	Pause timer register	32 bits	Read/write
0x12C – 0x1FC	Reserved			
MAC Registers				
0x200	MAC_REGS_DATA	Tri-Speed MAC registers data register	32 bits	Read/write
0x204	MAC_REGS_ADDR_RW	Tri-Speed MAC registers address and read/write control register	32 bits	Read/write
0x208 – 0x2FC	Reserved			
0x00	MODE_BYTE0 MODE_BYTE1	Mode register	8 bits 8 bits	Read/write
0x02	TX_RX_CTL_BYTE0 TX_RX_CTL_BYTE1	Transmit and receive control register	8 bits 8 bits	Read/write

Table 5: LatticeMico32 Tri-Speed MAC Register Descriptions (Continued)

Address [11:0] (Note: only [11:2] is decoded)	Register Name	Description	Size	Type
0x04	MAX_PKT_SIZE_BYTE0	Maximum packet size register	8 bits	Read/write
	MAX_PKT_SIZE_BYTE1		8 bits	
0x06	NULL_BYTE			
	NULL_BYTE			
0x08	IPG_VAL_BYTE0	Inter-packet gap register	8 bits	Read/write
	IPG_VAL_BYTE1		8 bits	
0x0A	MAC_ADDR_0_BYTE0	Tri-Speed MAC address register 0	8 bits	Read/write
	MAC_ADDR_0_BYTE1		8 bits	
0x0C	MAC_ADDR_1_BYTE0	Tri-Speed MAC address register 1	8 bits	Read/write
	MAC_ADDR_1_BYTE1		8 bits	
0x0E	MAC_ADDR_2_BYTE0	Tri-Speed MAC address register 2	8 bits	Read/write
	MAC_ADDR_2_BYTE1		8 bits	
0x10	NULL_BYTE		8 bits	Read/write
	NULL_BYTE		8 bits	
0x12	TX_RX_STS_BYTE0	Transmit and receive status	8 bits	Read only
	TX_RX_STS_BYTE1		8 bits	
0x14	GMII_MNG_CTL_BYTE0	GMII management interface control register	8 bits	Read/write
	GMII_MNG_CTL_BYTE1		8 bits	
0x16	GMII_MNG_DAT_BYTE0	GMII management data register	8 bits	Read/write
	GMII_MNG_DAT_BYTE1		8 bits	
0x18	NULL_BYTE			
	NULL_BYTE			
0x22	MLT_TAB_0_BYTE0	Multicast_table_0	8 bits	Read/write
	MLT_TAB_0_BYTE1		8 bits	
0x24	MLT_TAB_1_BYTE0	Multicast_table_1	8 bits	Read/write
	MLT_TAB_1_BYTE1		8 bits	
0x26	MLT_TAB_2_BYTE0	Multicast_table_2	8 bits	Read/write
	MLT_TAB_2_BYTE1		8 bits	
0x28	MLT_TAB_3_BYTE0	Multicast_table_3	8 bits	Read/write
	MLT_TAB_3_BYTE1		8 bits	
0x2A	MLT_TAB_4_BYTE0	Multicast_table_4	8 bits	Read/write
	MLT_TAB_4_BYTE1		8 bits	
0x2C	MLT_TAB_5_BYTE0	Multicast_table_5	8 bits	Read/write
	MLT_TAB_5_BYTE1		8 bits	

Table 5: LatticeMico32 Tri-Speed MAC Register Descriptions (Continued)

Address [11:0] (Note: only [11:2] is decoded)	Register Name	Description	Size	Type
0x2E	MLT_TAB_6_BYTE0	Multicast_table_6	8 bits	Read/write
	MLT_TAB_6_BYTE1		8 bits	
0x30	MLT_TAB_7_BYTE0	Multicast_table_7	8 bits	Read/write
	MLT_TAB_7_BYTE1		8 bits	
0x32	VLAN_TAG_BYTE0	VLAN tag length/type	8 bits	Read only
	VLAN_TAG_BYTE1		8 bits	
0x34	PAUS_OP_BYTE0	Pause_opcode register	8 bits	Read/write
	PAUS_OP_BYTE1		8 bits	
Receive Statistics Counters				
0x300	RX_PKT_IGNR_CNT	Receive packet ignored counter register	32 bits	Read only, rollover
0x304	RX_LEN_CHK_ERR_CNT	Receive length check error counter	32 bits	Read only, rollover
0x308	RX_LNG_FRM_CNT	Receive long frames counter register	32 bits	Read only, rollover
0x30C	RX_SHRT_FRM_CNT	Receive short frames counter register	32 bits	Read only, rollover
0x310	RX_IPG_VIOL_CNT	Receive IPG violations counter register	32 bits	Read only, rollover
0x314	RX_CRC_ERR_CNT	Receive CRC errors counter register	32 bits	Read only, rollover
0x318	RX_OK_PKT_CNT	Receive OK packets counter register	32 bits	Read only, rollover
0x31C	RX_CTL_FRM_CNT	Receive control frame counter register	32 bits	Read only, rollover
0x320	RX_PAUSE_FRM_CNT	Receive pause frame counter register	32 bits	Read only, rollover
0x324	RX_MULTICAST_CNT	Receive multicast frame counter register	32 bits	Read only, rollover
0x328	RX_BRDCAST_CNT	Receive broadcast frame counter register	32 bits	Read only, rollover
0x32C	RX_VLAN_TAG_CNT	Receive VLAN tagged frame counter register	32 bits	Read only, rollover

Table 5: LatticeMico32 Tri-Speed MAC Register Descriptions (Continued)

Address [11:0] (Note: only [11:2] is decoded)	Register Name	Description	Size	Type
0x330	RX_PRE_SHRINK_CNT	Receive preamble shrink counter register	32 bits	Read only, rollover
0x334	RX_DRIB_NIB_CNT	Receive dribble nibble counter register	32 bits	Read only, rollover
0x338	RX_UNSUP_OPCD_CNT	Receive unsupported opcode counter register	32 bits	Read only, rollover
0x33C	RX_BYTE_CNT	Receive byte counter register	32 bits	Read only, rollover
0x340 – 0x3FC		Reserved		
Transmit Statistics Counters				
0x400	TX_UNICAST_CNT	Transmit unicast frame counter register	32 bits	Read only, rollover
0x404	TX_PAUSE_FRM_CNT	Transmit pause frame counter register	32 bits	Read only, rollover
0x408	TX_MULTICAST_CNT	Transmit multicast frame counter register	32 bits	Read only, rollover
0x40C	TX_BRDCAST_CNT	Transmit broadcast frame counter register	32 bits	Read only, rollover
0x410	TX_VLAN_TAG_CNT	Transmit VLAN tagged frame counter register	32 bits	Read only, rollover
0x414	TX_BAD_FCS_CNT	Transmit BAD FCS frame counter register	32 bits	Read only, rollover
0x418	TX_JUMBO_CNT	Transmit jumbo frame counter register	32 bits	Read only, rollover
0x41C	TX_BYTE_CNT	Transmit byte counter register	32 bits	Read only, rollover
0x420 – 0x4FC		Reserved		

Register Bit Definitions

Receive Packet-Length FIFO Register

Receive packet-length FIFO register (RX_LEN_FIFO) – 0x000, default value = undefined, read only

The receive packet-length FIFO register is at fixed address 0x000. Its lower two bytes contain the length of the next packet in the receive data FIFO, in bytes. It's upper two bytes contain the Rx statistics vector bits [31:16] from the Tx MAC (see *IPUG51 - Tri-speed Ethernet Media Access Controller User's Guide*). For example, a value of 0x59 read from the lower two bytes of this register indicates that there is an 89-byte packet in the RX_DATA_FIFO that must be read. A value of 0x0080 read from the upper two bytes of this register indicates a packet received with no errors. This register is intended to be read only. After this register is read, the "RX_DATA_FIFO" must be read.

31	30	29	28	27	26	25	24
Rx_stat_ vec_31	Rx_stat_ vec_30	Rx_stat_ vec_29	Rx_stat_ vec_28	Rx_stat_ vec_27	Rx_stat_ vec_26	Rx_stat_ vec_25	Rx_stat_ vec_24

23	22	21	20	19	18	17	16
Rx_stat_ vec_23	Rx_stat_ vec_22	Rx_stat_ vec_21	Rx_stat_ vec_20	Rx_stat_ vec_19	Rx_stat_ vec_18	Rx_stat_ vec_17	Rx_stat_ vec_16

15	14	13	12	11	10	9	8
Len_bit15	Len_bit14	Len_bit13	Len_bit12	Len_bit11	Len_bit10	Len_bit9	Len_bit8

7	6	5	4	3	2	1	0
Len_bit7	Len_bit6	Len_bit5	Len_bit4	Len_bit3	Len_bit2	Len_bit1	Len_bit0

Bits	Mnemonic	Description
31	Rx_stat_vec_31	Long Frame
30	Rx_stat_vec_30	Short Frame
29	Rx_stat_vec_29	IPG Violation
28	Rx_stat_vec_28	Not Used – Reserved.
27	Rx_stat_vec_27	Carrier Event Previously Seen
26	Rx_stat_vec_26	Packet Ignored
25	Rx_stat_vec_25	CRC Error

Bits	Mnemonic	Description
24	Rx_stat_vec_24	Length Check Error
23	Rx_stat_vec_23	Receive OK
22	Rx_stat_vec_22	Multicast Address
21	Rx_stat_vec_21	Broadcast Address
20	Rx_stat_vec_20	Dribble Nibble
19	Rx_stat_vec_19	Unsupported Opcode
18	Rx_stat_vec_18	Control Frame
17	Rx_stat_vec_17	PAUSE Frame
16	Rx_stat_vec_16	VLAN Tag Detected
15:8	Len_byte1	High-order byte of length field
7:0	Len_byte0	Low-order byte of length field

Receive Packet-Data FIFO Register

Receive packet-data FIFO register (RX_DATA_FIFO) – 0x004, default value = undefined, read only

The receive packet-data FIFO register is at fixed address 0x004 and is intended to be read only. After reading the RX_LEN_FIFO register, 32-bit double words can be successively read from this register location.

Note

When Dual Wishbone interfaces are selected, this register is on interface B and interface A has a null at this location.

31	30	29	28	27	26	25	24
Rx_data Byte_3 Bit_7	Rx_data Byte_3 Bit_6	Rx_data Byte_3 Bit_5	Rx_data Byte_3 Bit_4	Rx_data Byte_3 Bit_3	Rx_data Byte_3 Bit_2	Rx_data Byte_3 Bit_1	Rx_data Byte_3 Bit_0

23	22	21	20	19	18	17	16
Rx_data Byte_2 Bit_7	Rx_data Byte_2 Bit_6	Rx_data Byte_2 Bit_5	Rx_data Byte_2 Bit_4	Rx_data Byte_2 Bit_3	Rx_data Byte_2 Bit_2	Rx_data Byte_2 Bit_1	Rx_data Byte_2 Bit_0

15	14	13	12	11	10	9	8
Rx_data Byte_1 Bit_7	Rx_data Byte_1 Bit_6	Rx_data Byte_1 Bit_5	Rx_data Byte_1 Bit_4	Rx_data Byte_1 Bit_3	Rx_data Byte_1 Bit_2	Rx_data Byte_1 Bit_1	Rx_data Byte_1 Bit_0

7	6	5	4	3	2	1	0
Rx_data Byte_0 Bit_7	Rx_data Byte_0 Bit_6	Rx_data Byte_0 Bit_5	Rx_data Byte_0 Bit_4	Rx_data Byte_0 Bit_3	Rx_data Byte_0 Bit_2	Rx_data Byte_0 Bit_1	Rx_data Byte_0 Bit_0

Bits	Mnemonic	Description
31:24	Rx_data_byte_3	Receive data byte received first
23:16	Rx_data_byte_2	Receive data byte received second
15:8	Rx_data_byte_1	Receive data byte received third
7:0	Rx_data_byte_0	Receive data byte received fourth

Transmit Packet-Length FIFO Register

Transmit packet-length FIFO register (TX_LEN_FIFO) – 0x008, default value = N/A, read/write

The transmit packet-length FIFO register is at fixed address 0x008 and is intended to be read/write. It is used by the logic between the transmit FIFO and the Tri-Speed MAC. It should be written with the length of the next packet in the transmit data FIFO, in bytes.

Note

An entire transmit packet should be first written to the TX_DATA_FIFO register before this register is written to, because the transmit FIFO extraction logic uses the occupancy of the transmit length FIFO as an indication of when to read the transmit data FIFO. For example, a value of 0x59 written to this register is used by downstream logic to extract a 89-byte packet from the TX_DATA_FIFO and send it to the Tri-Speed MAC.

31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8
Len_bit15	Len_bit14	Len_bit13	Len_bit12	Len_bit11	Len_bit10	Len_bit9	Len_bit8

7	6	5	4	3	2	1	0
Len_bit7	Len_bit6	Len_bit5	Len_bit4	Len_bit3	Len_bit2	Len_bit1	Len_bit0

Bits	Mnemonic	Description
31:24	Not_used	Reserved
23:16	Not_used	Reserved
15:8	Len_byte1	High-order byte of length field
7:0	Len_byte0	Low-order byte of length field

Transmit Packet-Data FIFO Register

Transmit packet-data FIFO register (RX_DATA_FIFO) – 0x00C, default value = N/A, write only

The transmit packet-data FIFO register is at fixed address 0x00C and is intended to be write only. After the TX_LEN_FIFO register is written to, the transmit FIFO extraction logic reads the length from the transmit length FIFO and then successively reads double words from this register location to send to the Tri-Speed MAC.

Note

When Dual Wishbone interfaces are selected, this register is on interface B and interface A has a null at this location.

31	30	29	28	27	26	25	24
Tx_data Byte_3 Bit_7	Tx_data Byte_3 Bit_6	Tx_data Byte_3 Bit_5	Tx_data Byte_3 Bit_4	Tx_data Byte_3 Bit_3	Tx_data Byte_3 Bit_2	Tx_data Byte_3 Bit_1	Tx_data Byte_3 Bit_0

23	22	21	20	19	18	17	16
Tx_data Byte_2 Bit_7	Tx_data Byte_2 Bit_6	Tx_data Byte_2 Bit_5	Tx_data Byte_2 Bit_4	Tx_data Byte_2 Bit_3	Tx_data Byte_2 Bit_2	Tx_data Byte_2 Bit_1	Tx_data Byte_2 Bit_0

15	14	13	12	11	10	9	8
Tx_data	Tx_data	Tx_data	Tx_data	Tx_data	Tx_data	Tx_data	Tx_data
Byte_1	Byte_1	Byte_1	Byte_1	Byte_1	Byte_1	Byte_1	Byte_1
Bit_7	Bit_6	Bit_5	Bit_4	Bit_3	Bit_2	Bit_1	Bit_0

7	6	5	4	3	2	1	0
Tx_data	Tx_data	Tx_data	Tx_data	Tx_data	Tx_data	Tx_data	Tx_data
Byte_0	Byte_0	Byte_0	Byte_0	Byte_0	Byte_0	Byte_0	Byte_0
Bit_7	Bit_6	Bit_5	Bit_4	Bit_3	Bit_2	Bit_1	Bit_0

Bits	Mnemonic	Description
31:24	Tx_data_byte_3	Transmit data byte transmitted first
23:16	Tx_data_byte_2	Transmit data byte transmitted second
15:8	Tx_data_byte_1	Transmit data byte transmitted third
7:0	Tx_data_byte_0	Transmit data byte transmitted fourth

Version/Identification Register

Version/Identification register (VERID) – 0x100, default value = {0x0001, ID}, read only

The version/identification register is a read-only register that contains the version and identification fields. The upper 16 bits are a hard-coded Lattice Semiconductor version number. The identification is a 16-bit, user-defined parameter for the wrapper. You can manually set the wrapper identification in the wb_wrapper.v file with the defparam above the ctl_status_regs instance (defparam ctl_status_regs.ID = 16'd0).

31	30	29	28	27	26	25	24
Ver_bit15	Ver_bit14	Ver_bit13	Ver_bit12	Ver_bit11	Ver_bit10	Ver_bit9	Ver_bit8

23	22	21	20	19	18	17	16
Ver_bit7	Ver_bit6	Ver_bit5	Ver_bit4	Ver_bit3	Ver_bit2	Ver_bit1	Ver_bit0

15	14	13	12	11	10	9	8
ID_bit15	ID_bit14	ID_bit13	ID_bit12	ID_bit11	ID_bit10	ID_bit9	ID_bit8

7	6	5	4	3	2	1	0
ID_bit7	ID_bit6	ID_bit5	ID_bit4	ID_bit3	ID_bit2	ID_bit1	ID_bit0

Bits	Mnemonic	Description
31:24	Version_byte1	Hard-coded Lattice Semiconductor version number set to 0x00
23:16	Version_byte0	Hard-coded Lattice Semiconductor version number set to 0x01
15:8	ID_byte1	Identification bits that you set by using the ID_PARM parameter
7:0	ID_byte0	Identification bits that you set by using the ID_PARM parameter

Interrupt Source Register

Interrupt source register (INTR_SRC) – 0x104, default value = 0x0000_0000, read only, clear on read

The interrupt source register contains the receive and transmit interrupt sources. It is a read-only register, which is cleared on a read. The Tri-Speed MAC performs a logical AND on the interrupt bits from this register and the enable bits in the interrupt enable register. See “Interrupt Hierarchy” on page 46 for more details.

31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16
						RX_smry	TX_smry

15	14	13	12	11	10	9	8
			Rx_fifo_full	Rx_error	Rx_fifo_error	Rx_fifo_ almost_full	Rx_pkt_rdy

7	6	5	4	3	2	1	0
			Tx_fifo_full	Tx_discfrm	Tx_fifo_ almost_ empty	Tx_fifo_ almost_full	Tx_pkt_sent

Bits	Mnemonic	Description
31:24	Not_used	Reserved
23:18	Not_used	Reserved
17	Rx_smry	Receive interrupt source summary bit
16	Tx_smry	Transmit interrupt source summary bit
15:13	Not_used	Reserved
12	Rx_fifo_full	Receive FIFO is full (error condition)
11	Rx_error	Tri-Speed MAC detected receive packet error
10	Rx_fifo_error	Receive FIFO error condition
9	Rx_fifo_almost_full	Receive FIFO is almost full (based on threshold set)
8	Rx_pkt_rdy	Receive FIFO is ready (completely loaded in receive FIFO)
7:5	Not_used	Reserved
4	Tx_fifo_full	Transmit FIFO is full (error condition)
3	Tx_discrm	Tri-Speed MAC detected a transmit packet error
2	Tx_fifo_almost_empty	Transmit FIFO is almost empty (based on threshold set)
1	Tx_fifo_almost_full	Transmit FIFO is almost full (based on threshold set)
0	Tx_pkt_sent	Transmit packet sent to Tri-Speed MAC indication

Interrupt Enable Register

Interrupt enable register (INTR_ENB) – 0x108, default value = 0x0000_0000, read/write

The interrupt enable register is used in conjunction with the interrupt source register. The Tri-Speed MAC performs a logical AND on the interrupt bits from this register and the bits from the interrupt source register. See “Interrupt Hierarchy” on page 46 for more details. This register is read/write.

31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16
					Intr_enb	RX_smry_ enb	TX_smry_ enb

15	14	13	12	11	10	9	8
			Enb_12	Enb_11	Enb_10	Enb_9	Enb_8

7	6	5	4	3	2	1	0
			Enb_4	Enb_3	Enb_2	Enb_1	Enb_0

Bits	Mnemonic	Description
31:24	Not_used	Reserved
23:19	Not_used	Reserved
18	Intr_enb	Interrupt enable
17	Rx_smry_enb	Receive interrupt source summary bit enable
16	Tx_smry_enb	Transmit interrupt source summary bit enable
15:13	Not_used	Reserved
12	Rx_fifo_full	Interrupt enable
11	Rx_error	Interrupt enable
10	Rx_fifo_error	Interrupt enable
9	Rx_fifo_almost_full	Interrupt enable
8	Rx_pkt_rdy	Interrupt enable
7:5	Not_used	Reserved
4	Tx_fifo_full	Interrupt enable
3	Tx_discfrm	Interrupt enable
2	Tx_fifo_almost_empty	Interrupt enable
1	Tx_fifo_almost_full	Interrupt enable
0	Tx_pkt_sent	Interrupt enable

Receive Status Register

Receive status register (RX_STATUS) – 0x10C, default value = 0x0000_0000, read only, clear on read

The receive status register contains the receive status bits. It is a mirror of the receive portion of the interrupt source register and can be used for software

polling and so forth without affecting the interrupt source register. This register is a read-only register, which is cleared on a read.

31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8

7	6	5	4	3	2	1	0
			Rx_fifo_full	Rx_error	Rx_fifo_error	Rx_fifo_ almost_full	Rx_pkt_rdy

Bits	Mnemonic	Description
31:5	Not_used	Reserved
4	Rx_fifo_full	Receive FIFO is full (error condition)
3	Rx_error	Tri-Speed MAC detected receive packet error
2	Rx_fifo_error	Receive FIFO error condition
1	Rx_fifo_almost_full	Receive FIFO is almost full (based on threshold set)
0	Rx_pkt_rdy	Receive packet is ready (completely loaded in receive FIFO)

Transmit Status Register

Transmit status register (TX_STATUS) – 0x110, default value = 0x0000_0000, read only, clear on read

The transmit status register contains the transmit status bits. It is a mirror of the transmit portion of the interrupt source register and can be used for software polling and so forth without affecting the interrupt source register. This register is a read-only register, which is cleared on a read.

31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8

7	6	5	4	3	2	1	0
			Tx_fifo_full	Tx_discfrm	Tx_fifo_ almost_ empty	Tx_fifo_ almost_full	Tx_pkt_rdy

Bits	Mnemonic	Description
31:5	Not_used	Reserved
4	Tx_fifo_full	Transmit FIFO is full (error condition)
3	Tx_discfrm	Tri-Speed MAC detected transmit packet error
2	Tx_fifo_almost_empty	Transmit FIFO is almost empty (based on threshold set)
1	Tx_fifo_almost_full	Transmit FIFO is almost full (based on threshold set)
0	Tx_pkt_sent	Transmit packet sent to Tri-Speed MAC indication

Receive Frames Count Register

Receive frames count register (RX_FRAMES_CNT) – 0x114, default value = 0x0000_0000, read only

The receive frames count register contains the number of frames currently in the receive data FIFO. The RX_FRAMES_CNT register is incremented by one each time the receive-length FIFO is written to and decremented by one each time the receive-length FIFO is read. This register is read only.

31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8

7	6	5	4	3	2	1	0
Rxfrm_cnt7	Rxfrm_cnt6	Rxfrm_cnt5	Rxfrm_cnt4	Rxfrm_cnt3	Rxfrm_cnt2	Rxfrm_cnt1	Rxfrm_cnt0

Bits	Mnemonic	Description
31:24	Not_used	Reserved
23:16	Not_used	Reserved
15:8	Not_used	Reserved
7:0	Rxfrm_cnt	Receive frame counter bits 7:0 - count valid from 0 to 255

Transmit Frames Count Register

Transmit frames count register (TX_FRAMES_CNT) – 0x118, default value = 0x0000_0000, read only

The transmit frames count register contains the number of frames currently in the transmit data FIFO. The TX_FRAMES_CNT register is incremented by one each time the transmit length FIFO is written to and decremented by one each time a transmit packet is transmitted by the Tri-Speed MAC (transmit-length FIFO is read and packet sent). This register is read only.

31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8

7	6	5	4	3	2	1	0
Txfrm_cnt7	Txfrm_cnt6	Txfrm_cnt5	Txfrm_cnt4	Txfrm_cnt3	Txfrm_cnt2	Txfrm_cnt1	Txfrm_cnt0

Bits	Mnemonic	Description
31:24	Not_used	Reserved
23:16	Not_used	Reserved
15:8	Not_used	Reserved
7:0	Txfrm_cnt	Transmit frame counter bits 7:0 - count valid from 0 to 255

Receive FIFO Threshold Register

Receive FIFO threshold register (RX_FIFO_TH) – 0x11C, default value = 0x0000_0000, read/write

The receive FIFO threshold register is a read/write register used to set the almost empty and almost full threshold watermarks for the receive-data FIFO. The receive-data FIFO is composed of four cascaded 512x32 FIFOs to effectively build a 2048x32 FIFO (8192 bytes). The almost full and almost empty thresholds, therefore, are set on the basis of 512 locations for the top and bottom of the four cascaded FIFOs. A 9-bit value represents these watermarks.

31	30	29	28	27	26	25	24
							AE_TH_b8

23	22	21	20	19	18	17	16
AE_TH_b7	AE_TH_b6	AE_TH_b5	AE_TH_b4	AE_TH_b3	AE_TH_b2	AE_TH_b1	AE_TH_b0

15	14	13	12	11	10	9	8
							AF_TH_b8

7	6	5	4	3	2	1	0
AF_TH_b7	AF_TH_b6	AF_TH_b5	AF_TH_b4	AF_TH_b3	AF_TH_b2	AF_TH_b1	AF_TH_b0

Bits	Mnemonic	Description
31:25	Not_used	Reserved
24:16	AE[8:0]	RX_DATA FIFO almost empty watermark bits
15:9	Not_used	Reserved
8:0	AF[8:0]	RX_DATA FIFO almost full watermark bits

Transmit FIFO Threshold Register

Transmit FIFO threshold register (TX_FIFO_TH) – 0x120, default value = 0x0000_0000, read/write

The transmit FIFO threshold register is a read/write register used to set the almost empty and almost full threshold watermarks for the transmit-data FIFO. The transmit-data FIFO is composed of four cascaded 512x32 FIFOs to effectively build a 2048x32 FIFO (8192 bytes). The almost full and almost empty thresholds, therefore, are set on the basis of 512 locations for the top and bottom of the four cascaded FIFOs. A 9-bit value represents these watermarks.

31	30	29	28	27	26	25	24
							AE_TH_b8

23	22	21	20	19	18	17	16
AE_TH_b7	AE_TH_b6	AE_TH_b5	AE_TH_b4	AE_TH_b3	AE_TH_b2	AE_TH_b1	AE_TH_b0

15	14	13	12	11	10	9	8
							AF_TH_b8

7	6	5	4	3	2	1	0
AF_TH_b7	AF_TH_b6	AF_TH_b5	AF_TH_b4	AF_TH_b3	AF_TH_b2	AF_TH_b1	AF_TH_b0

Bits	Mnemonic	Description
31:25	Not_used	Reserved
24:16	AE[8:0]	TX_DATA FIFO almost empty watermark bits
15:9	Not_used	Reserved
8:0	AF[8:0]	TX_DATA FIFO almost full watermark bits

System Control Register

System control register (SYS_CTL) – 0x124, default value = 0x0000_0000, read/write

The system control register is a read/write register used to control the Tri-Speed MAC and receive and transmit FIFOs for system-related tasks.

31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8

7	6	5	4	3	2	1	0
Drop_err_pkts	Addr_Swap_enable	Loop_back_enable	Tx_fifo_flush	Rx_fifo_flush	Tx_sndpausreq	TX_fifoctrl	Ignore_next_pkt

Bits	Mnemonic	Description
31:8	Not_used	Reserved
7	Drop_err_pkts	Drops error packets during Loop back
6	Addr_swap_enable	Enables swapping of the DEST address and the SRC address during Loop back
5	Loop_back_enable	Enables Loop Back
4	Tx_fifo_flush	Flushes the transmit FIFO
3	Rx_fifo_flush	Flushes the receive FIFO
2	Tx_sndpausreq	When the pause frame request is asserted, the Tri-Speed MAC transmits a pause frame. This is also the qualifying signal for the tx_sndpausetim bus.
1	Tx_fifoctrl	The FIFO control frame signal indicates whether the current frame in the transmit FIFO is a control frame or a data frame. The following values apply: ◆ 1 = Control frame ◆ 0 = Normal frame
0	Ignore_next_pkt	The ignore next packet bit should be asserted by the host to prevent a receive FIFO full condition. The receive Tri-Speed MAC continues dropping packets as long as this bit is asserted. This bit can be set asynchronously to the Tri-Speed MAC operation.

Pause Timer Register

Pause timer register (PAUSE_TMR) – 0x128, default value = 0x0000_0000, read/write

The pause timer register is a read/write register used to set the pause time value that should be sent in the pause frame sent by the Tri-Speed MAC. Two different methods are used for transmitting a pause frame:

- The application layer can form a pause frame and submit it for transmission through the FIFO.
- The application layer signals the transmit Tri-Speed MAC directly to transmit a pause frame by asserting the tx_sndpausreq bit in the sys_ctl register. In this case, the transmit Tri-Speed MAC completes transmission of the current packet and transmits a pause frame with the pause time value supplied through the value in this pause_tmr register.

31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8
PT_bit15	PT_bit14	PT_bit13	PT_bit12	PT_bit11	PT_bit10	PT_bit9	PT_bit8

7	6	5	4	3	2	1	0
PT_bit7	PT_bit6	PT_bit5	PT_bit4	PT_bit3	PT_bit2	PT_bit1	PT_bit0

Bits	Mnemonic	Description
31:16	Not_used	Reserved
15:10	PT[15:0]	Pause timer value

Tri-Speed MAC Registers Data Register

Tri-Speed MAC registers data register (MAC_REGS_DATA) – 0x200, default value = 0x0000_0000, read/write

The Tri-Speed MAC registers data register is a read/write register that contains the data read back or the data to write to at the address specified in MAC_REG_ADDR.

31	30	29	28	27	26	25	24

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8
BYTE1_7	BYTE1_6	BYTE1_5	BYTE1_4	BYTE1_3	BYTE1_2	BYTE1_1	BYTE1_0

7	6	5	4	3	2	1	0
BYTE0_7	BYTE0_6	BYTE0_5	BYTE0_4	BYTE0_3	BYTE0_2	BYTE0_1	BYTE0_0

Bits	Mnemonic	Description
31:16	Not_used	
15:8	Data_high_byte	Read from or write to high-byte data Tri-Speed MAC registers specified by the address in MAC_REG_ADDR
7:0	Data_low_byte	Read from or write to low-byte data Tri-Speed MAC registers specified by the address in MAC_REG_ADDR

Tri-Speed MAC Registers Address and Read/ Write Control Register

Tri-Speed MAC registers address and read/write control register (MAC_REGS_ADDR_RW) – 0x204, default value = 0x0000_0000, read/write

The Tri-Speed MAC registers address is a read/write register that contains the read or write address of the Tri-Speed MAC register to be accessed and a read/write control bit that indicates a read (low) or write (high) operation.

Note

The address written to this register is on word boundaries.

31	30	29	28	27	26	25	24
RW							

23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8

7	6	5	4	3	2	1	0
Add_7	Add_6	Add_b5	Add_4	Add_3	Add_2	Add_1	Add_0

Bits	Mnemonic	Description
31	Read/write bit	Control bit that indicates a read (low) or write (high) operation.
30:8	Unused	
7:0	Address	8-bit Tri-Speed MAC address

Timing Diagrams

Figure 5 shows the Tri-Speed MAC reading an 89-byte receive packet using Classic Wishbone cycles.

Figure 5: Tri-Speed MAC Reading an 89-Byte Rx Packet Using Classic Cycles

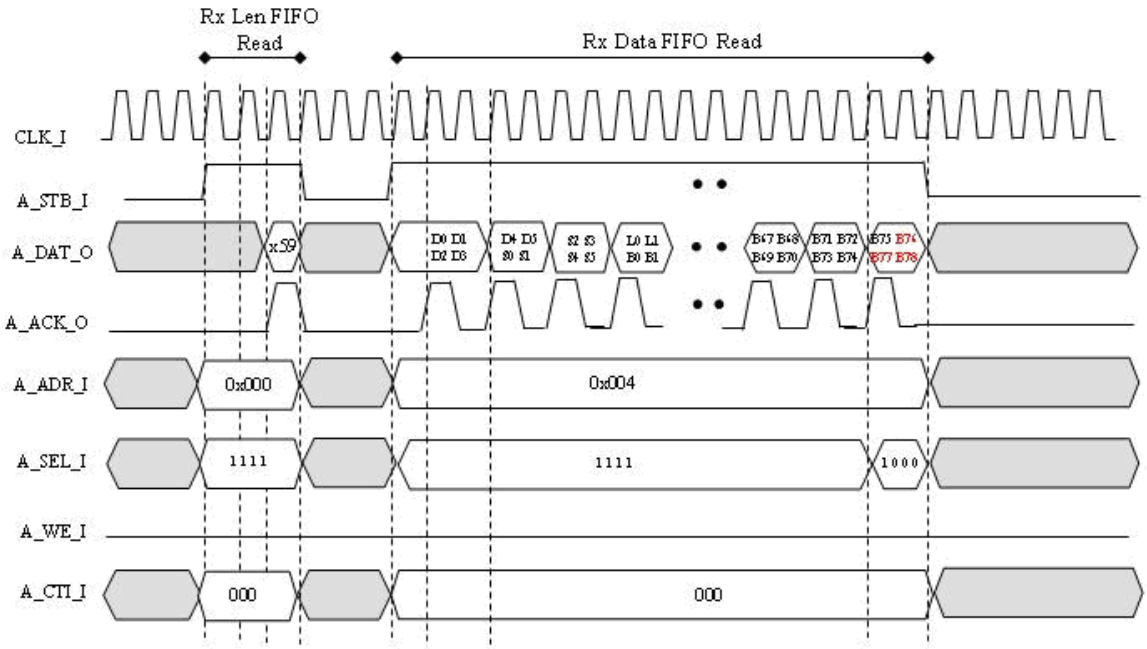


Figure 6 shows the Tri-Speed MAC reading an an 89-byte Rx Packet using Constant Address Burst Wishbone cycles.

Figure 6: Tri-Speed MAC Reading an 89-Byte Rx Packet Using Constant Address Burst Cycles

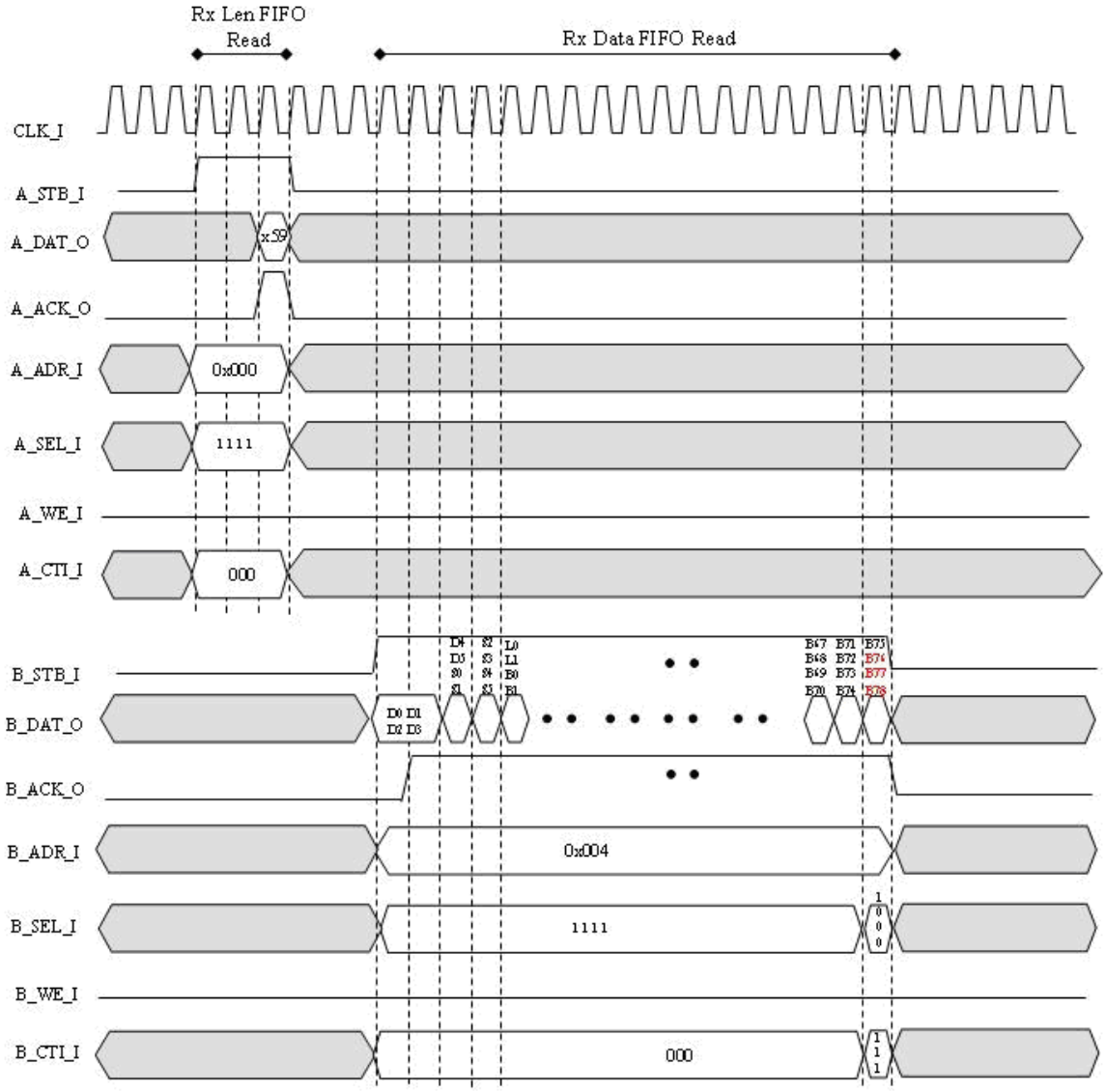


Figure 7 shows the Tri-Speed MAC writing an 80-byte Tx Packet Using Classic Wishbone cycles.

Figure 7: Tri-Speed MAC Writing an 80-byte Tx Packet Using Classic Wishbone cycles

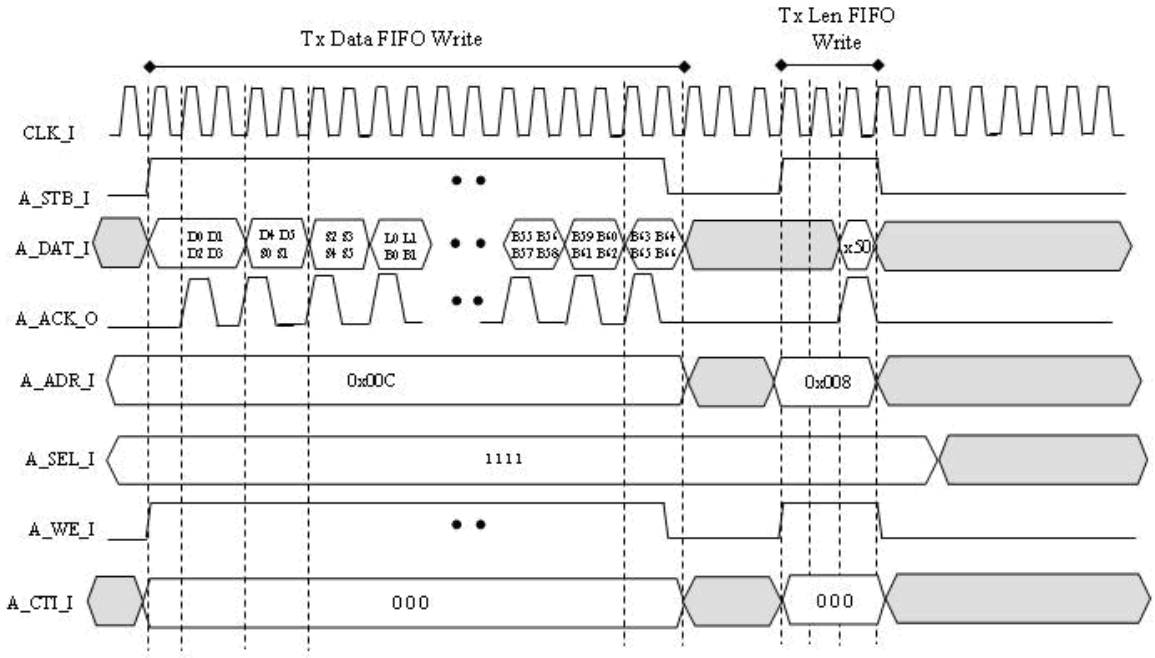


Figure 8 shows the Tri-Speed MAC writing an 80-byte Tx Packet using Constant Address Burst Wishbone cycles.

Figure 8: Tri-Speed MAC Writing an 80-byte Tx Packing Using Constant Address Burst Cycles

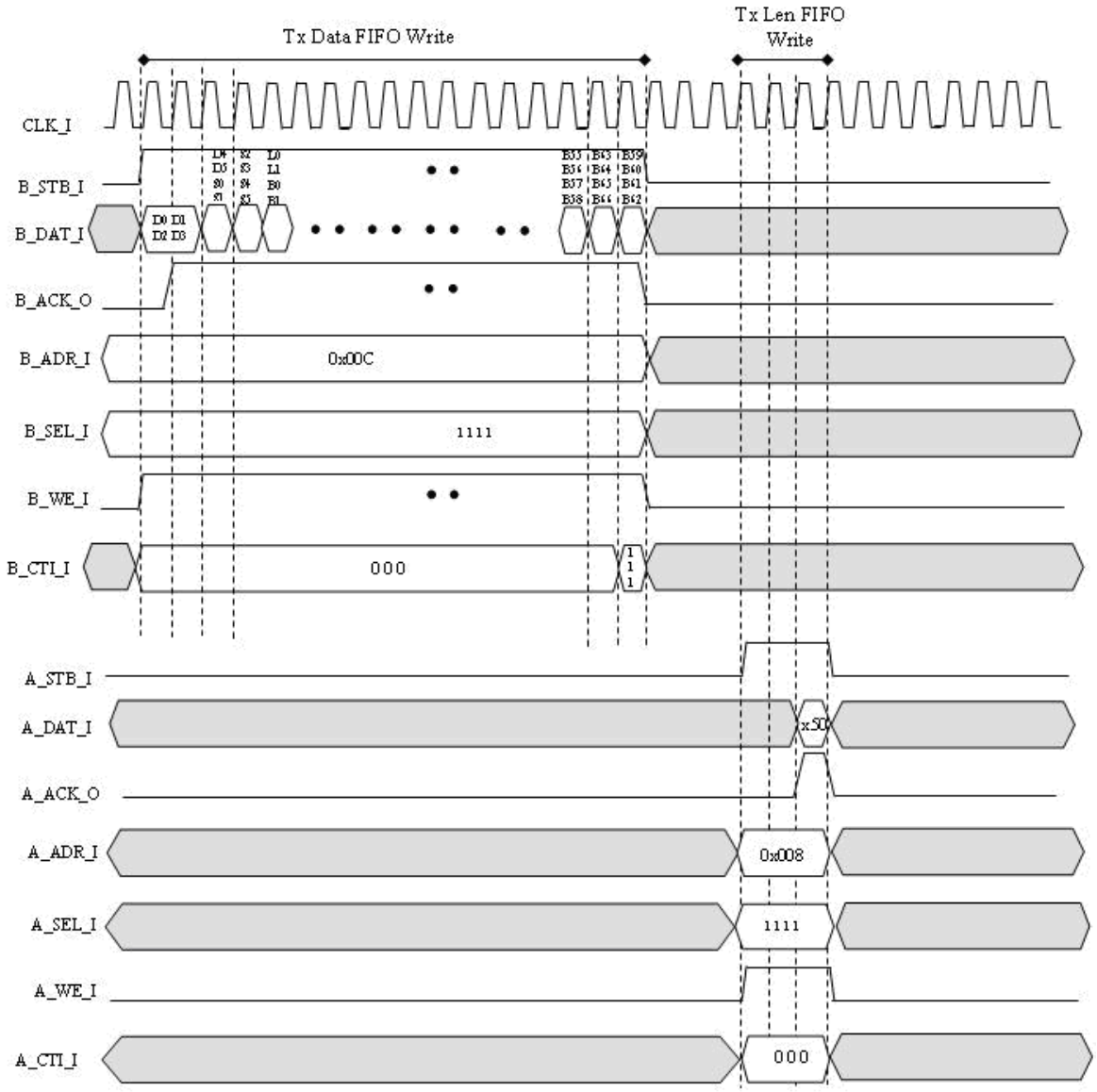
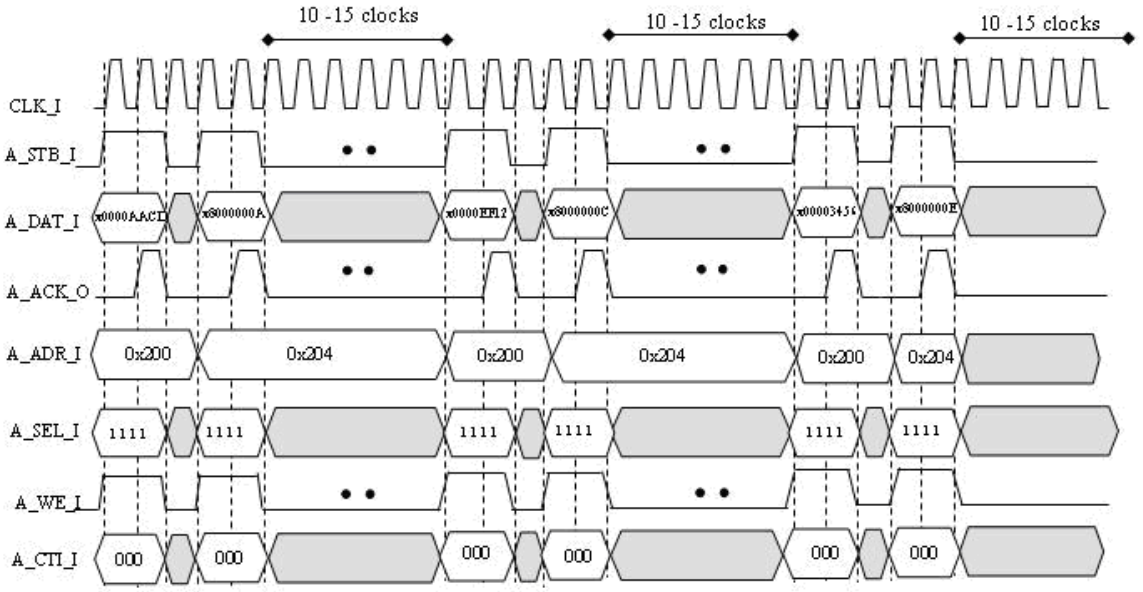


Figure 9 shows the Tri-Speec MAC writing MAC Address to AA-CD-EF-12-34-56 using Classic Wishbone cycles.

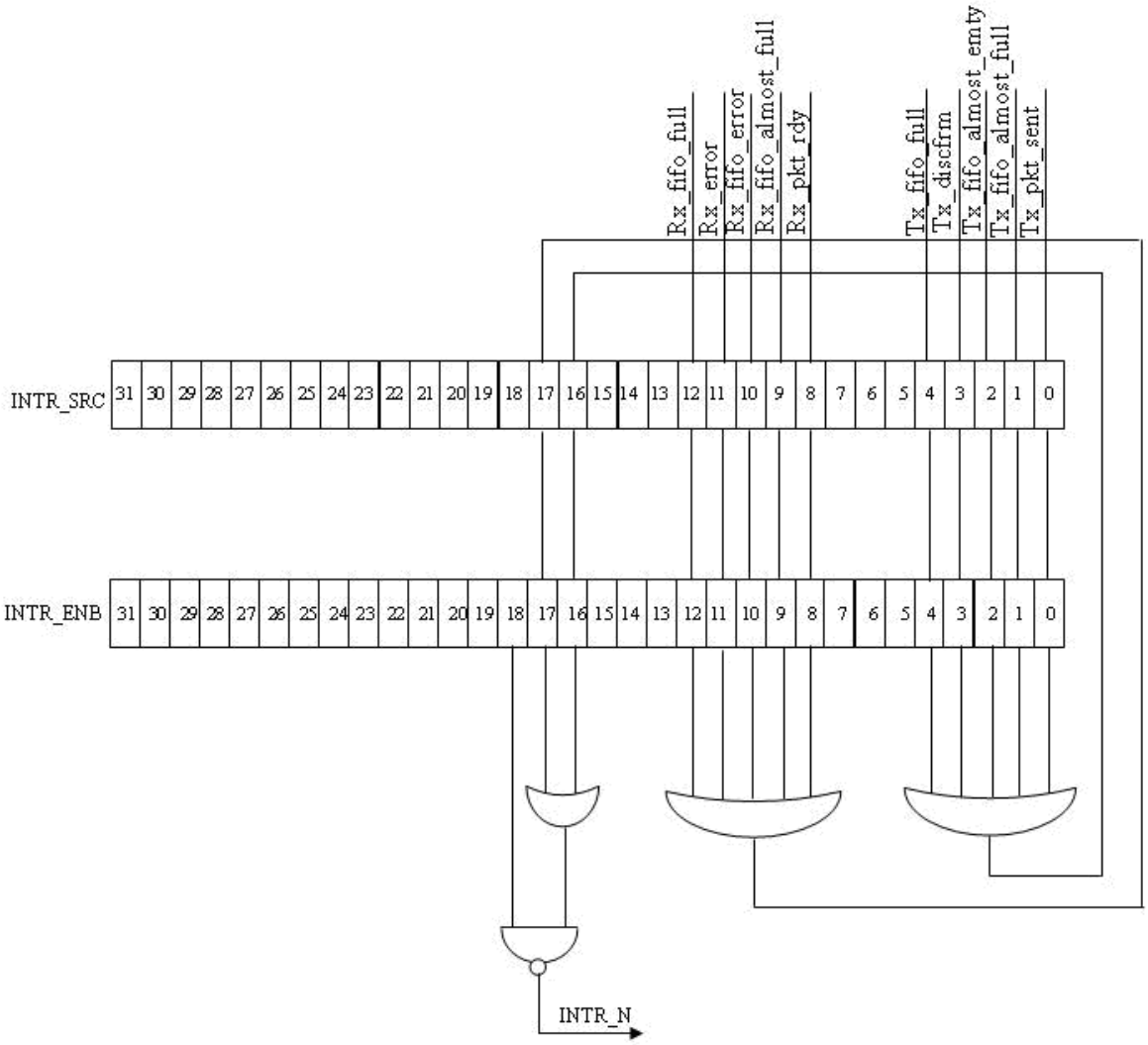
Figure 9: Tri-Speed MAC writing MAC Address using Classic Wishbone cycles



Interrupt Hierarchy

Figure 10 shows the interrupt hierarchy used.

Figure 10: Tri-Speed MAC Interrupt Hierarchy



EBR Resource Utilization

The number of EBRs in the LatticeMico32 Tri-Speed Ethernet MAC WISHBONE wrapper depends on the depth of the Tx and Rx FIFOs.

- If the depth of the Tx and Rx FIFOs is 512, the Tri-Speed Ethernet MAC uses 6 EBRs.
- If the depth of the Tx and Rx FIFOs is 1024, the Tri-Speed Ethernet MAC uses 8 EBRs.

- If the depth of the Tx and Rx FIFOs is 2048, the Tri-Speed Ethernet MAC uses 12 EBRs.

For information on the number of EBRs in the LatticeMico32 Tri-Speed Ethernet MAC IP core, see the *Tri-Speed Ethernet Media Access Controller* data sheet.

Additional Information

Following is some additional information about the LatticeMico32 Tri-Speed MAC:

- It is assumed that any data structures used by the TCP/IP layer, like linked lists and their maintenance, is handled by the LatticeMico32 software.
- It is assumed that the additional processor clock cycles involved in reading and writing control registers through the register control path and any additional software processing does not adversely impact the throughput requirements.
- The depth of the FIFO buffers is configurable to 2, 4, or 8 kilobytes.
- The full set of IEEE 802.3 statistics counters and registers is not supported. A subset of the statistics counters is implemented. You can easily remove this module with the Include Statistics Regs option in the graphical user interface.

Reference Information

The following documents provide more information on the Tri-Speed MAC:

- *IPUG51 - Tri-Speed Ethernet Media Access Controller User's Guide*
- *IEEE 802.3 Ethernet Specifications for 10/100/1G*

You can also obtain more information at the following Web site:

http://www.latticesemi.com/products/intellectualproperty/ipcores/mico32/