

Features

- **Compliant to IEEE 802.3z Standard**
- **Generic Host Interface**
 - Configurable 8-bit or 16-bit and Greater Data Bus
- **16-bit Wide Internal Data Path**
- **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
 - Optional FCS Transmission and Reception
 - Optional MII Management Interface Module

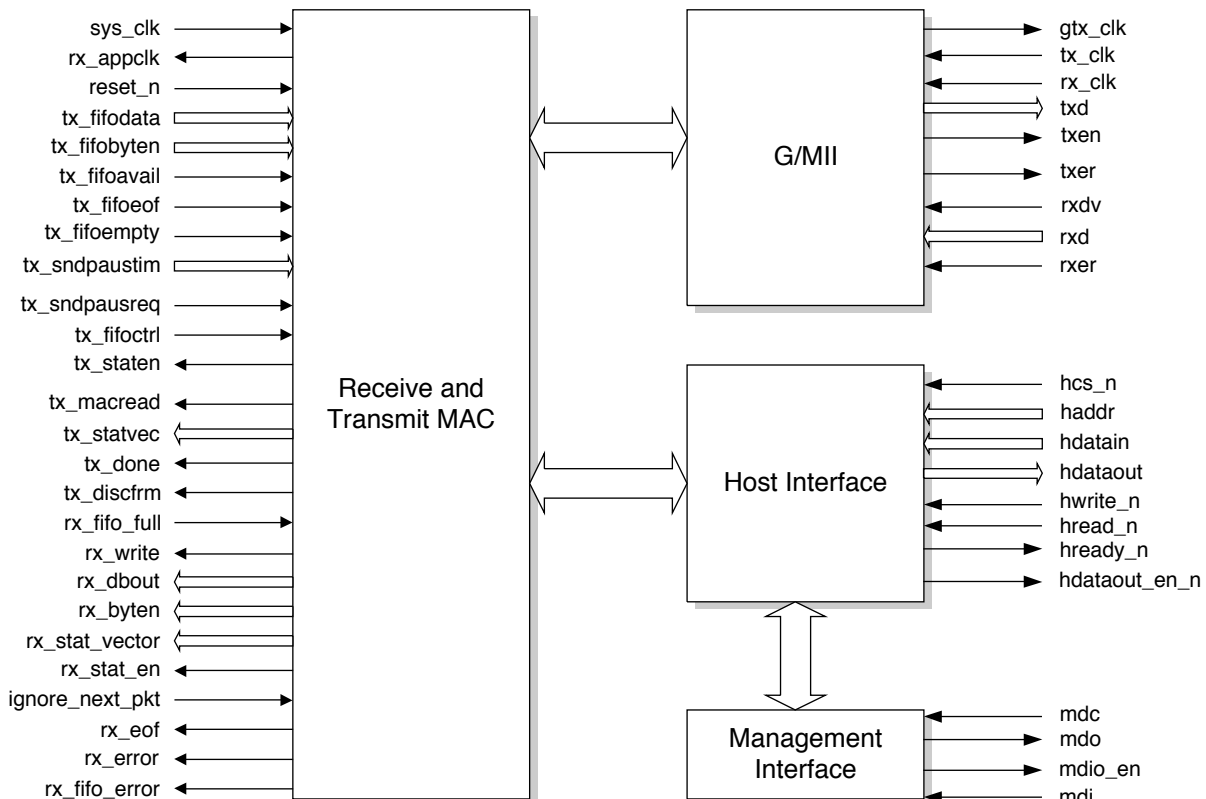
General Description

The Triple Speed Ethernet Media Access Controller (Tri-Speed MAC) core can 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 in either one of these modes. The netlist cannot auto-negotiate between the two different modes.

The Tri-Speed MAC transmits and receives data between a host processor and an Ethernet network. The main function of the Tri-Speed 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 Tri-Speed MAC extracts the different components of a frame and transfers them to higher applications through the FIFO interface.

Block Diagram

Figure 1. Simplified Block Diagram - Tri-Speed Ethernet MAC



The data received from the G/MII interface is first buffered until sufficient data is available to be processed by the Receive MAC. The Receive MAC checks the address of the received packet and validates whether the frame can be received before transferring it onto the FIFO. The preamble and the Start of Frame Delimiter (SFD) information are then extracted from the incoming frame while the rest of the frame is transferred into the FIFO. This feature has the following two benefits. The systems do not need to re-calculate the 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 if the frame was received error-free.

The transmitting side is explained in the Transmit MAC section.

Figure 2. Untagged Ethernet Frame Format

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

Functional Description

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

The block diagram of the Tri-Speed MAC core is shown in Figure 1. 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 de-asserted. The address of the register that will be written into is placed on the `haddr` bus with the valid data being placed onto the `hdatain` bus. The contents of the address and data bus should remain unchanged until the Tri-Speed MAC core asserts the `hready_n` signal. The `hcs_n` and `hwrite_n` signals need to be asserted for only one cycle to perform a single write operation. 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 for one cycle with the `hwrite_n` signal remaining de-asserted. The address of the register that needs to be read is placed on the `haddr` bus. The Tri-Speed MAC places the contents of the register being read onto the `hdataout` bus and qualifies it with the assertion of the `hready_n` signal. The `haddr` bus should remain un-changed until the `hready_n` signal is asserted.

Receive MAC

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

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

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

The entire frame, except the preamble and SFD bytes, are sent to the FIFO by default. 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.

Transmit MAC

The Transmit MAC is responsible for controlling access to the medium. The Transmit MAC accepts the data from the transmit FIFO, formats it into an Ethernet packet and passes it over to the G/MII module. Once the FIFO indicates that enough data is available for transmission, the Transmit MAC reads the FIFO. The Transmit MAC derives the DA, SA, L/T, and the DATA fields from higher applications through the FIFO interface. Then it encapsulates them into an un-tagged Ethernet frame as shown in Figure 2 before transmitting. This frame, however, is not sent over the network until the network has been idle for a minimum of the 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, which is FCS. If padding is not disabled, all short frames are padded with pad symbols.

Once the transmission ends, a `tx_stat_vector` bus is presented to the host that includes all the statistical information collected in the process of transmitting the frame. Assertion of the `tx_stat_en` signal qualifies this bus.

If a FIFO underflow occurs, the Transmit MAC inserts erroneous CRC bits into the packet to guarantee that the receiver detects the error in the packet.

In the full duplex mode, filling up the receiver's buffer rapidly is possible. In such cases, the receiver sends flow control (PAUSE) frames to the transmitter, requesting it to stop transmitting frames until it can free the buffers. Then the transmitter completes sending the current frame and stops for the duration specified in the PAUSE frame.

Internal Buffer and FIFO Interfaces

In the 10/100 mode, the Transmit and Receive sections each contain FIFO for data buffering. On the transmitting side, the internal buffer stores the first 64 bytes of the frame. This ensures that the Tri-Speed MAC can re-transmit the frame automatically without help from the application software during an in-window collision. This important feature prevents the propagation of collision information into the application software.

The 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 path has an internal buffer to support this feature.

G/MII

The G/MII module interfaces to the PHY device. The only module in the core that is speed dependent is the G/MII module. It relies on the clock supplied by the external PHY. The speed-dependent G/MII interface converts the data for the eight-bit wide G/MII data bus.

Although not designated as a submodule, 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 in the 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 into or reads from the PHY registers. A single MIIM can address up to 32 PHY devices; as a result, the MIIM module is user-selectable during synthesis. 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 10MHz `mdc` clock. Therefore, the Tri-Speed MAC can have a MIIM that is capable of running at up to 10 MHz.

The read or write operations are specified in the `GMII_MNG_CTL` register along with the PHY that is addressed and the register within the PHY that needs to be accessed. For a write operation, the data to be written is stored in the `GMII_MNG_DAT` register. In case of 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 eight-bit host interfaces, the individual bytes of the registers are accessed through their corresponding address with the lower address pointing to the lower byte.
- In 16-bit or greater host interfaces, only even numbered addresses should be used.
- All registers except the `MODE` register can be written into only when the core is in the `IDLE` state, as defined in the `MAC_INT_STS` register. The `MODE` register is the only register that can be written into while the core is enabled.

Table 1 lists the Tri-Speed MAC registers accessible via the Host Interface.

Table 1. Tri-Speed MAC Internal Registers

Register Description	Mnemonic	I/O Address
Mode Register	MODE	00H - 01H
Transmit and Receive Control Register	TX_RX_CTL	02H - 03H
Maximum Packet Size Register	MAX_PKT_SIZE	04H - 05H
Inter-Packet Gap Register	IPG_VAL	08H - 09H
Tri-Speed MAC Address Register 0	MAC_ADDR_0	0AH - 0BH
Tri-Speed MAC Address Register 1	MAC_ADDR_1	0CH - 0DH
Tri-Speed MAC Address Register 2	MAC_ADDR_2	0EH - 0FH
Transmit and Receive Status	TX_RX_STS	12H - 13H
GMII Management Interface Control Register	GMII_MNG_CTL	14H - 15H
GMII Management Data Register	GMII_MNG_DAT	16H - 17H
VLAN Tag Length/Type Register	VLAN_TAG	32H - 33H
Multicast_table_0	MLT_TAB_0	22H - 23H

Table 1. Tri-Speed MAC Internal Registers (Continued)

Register Description	Mnemonic	I/O Address
Multicast_table_1	MLT_TAB_1	24H - 25H
Multicast_table_2	MLT_TAB_2	26H - 27H
Multicast_table_3	MLT_TAB_3	28H - 29H
Multicast_table_4	MLT_TAB_4	2AH - 2BH
Multicast_table_5	MLT_TAB_5	2CH - 2DH
Multicast_table_6	MLT_TAB_6	2EH - 2FH
Multicast_table_7	MLT_TAB_7	30H - 31H
Pause_opcode	PAUS_OP	34H - 35H

Parameter Descriptions

The Tri-Speed MAC Core includes configurable parameters that allow integration with the user's application. Figure 2 shows the configurable parameters.

Table 2. Tri-Speed MAC Configuration Parameters

Parameter	Value Range	Default	Description
MODE	1000 Mbps or 10/100 Mbps	1000 Mbps Mode	This parameter determines the Ethernet speed that the core can access.
MIIM_MODULE	Include or Do Not Include	Include	This parameter determines if the optional MIIM module is included in the core's implementation.
CPU_DATA_WIDTH	8 or 16 bits	16 bits	This parameter determines the data bus width of the bus that communicates with the host.

Signal Descriptions

Table 3. Tri-Speed MAC Input and Output Signals

Signal	Type	Description
Clocks and Reset		
sys_clk	Input	System Clock. This is used to clock the host interface. The Tx MAC is clocked by this signal when operating in the gigabit mode. All the input and the output signals of the Tx MAC are synchronous to this clock in the gigabit mode. Its frequency is always 62.5 MHz.
rx_appclk	Output	Rx 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	Gigabit Transmit Clock. This clock is used in the gigabit mode only. The transmit signals that are output on the GMII interface are synchronous to this clock. This clock has a frequency of 125 MHz.
tx_clk	Input	Transmit Clock. This clock is used in the 10/100 Mbps mode only where all the signals are at the MII. The Tx MAC application interface are synchronous to this signal. The Tx MAC is also clocked on this signal in the 10/100 Mbps mode. This clock has a frequency of 2.5/25 MHz.
rx_clk	Input	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. It is provided un-modified to the Receive MAC section. The receive signals at the GMII interface are always synchronous to rx_clk.
mdc	Input	Management Data Clock. This clock is used only when the Management Interface module is implemented.
reset_n	Input	Reset. This is an active low signal that resets the internal registers and internal logic. The I/O signals are driven to their inactive levels.

Table 3. Tri-Speed MAC Input and Output Signals (Continued)

Signal	Type	Description
Host Interface		
hcs_n	Input	Chip Select. This is an active low signal used to select the core.
haddr[(addr_width-1):0]	Input	Address. This selects one of the internal registers
hdatain[(data_width-1):0]	Input	Data Bus Input. The CPU writes into the internal registers through the data bus.
hwrite_n	Input	IO Register Write. This active low signal is used to write data to the selected register.
hread_n	Input	IO Register Read. This active low signal is used to read data from the selected register.
hready_n	Output	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 is ready on data bus to be driven out.
hdataout_en_n	Output	Data out Enable. This signal is driven low whenever the core outputs valid data onto the hdataout bus. This signal can be used to build a bi-directional bus.
hdataout[(data_width-1):0]	Output	Data Bus Output: The CPU reads the internal registers through the data bus.
Transmit MAC Application Interface		
tx_fifodata[15:0]	Input	Transmit FIFO read data. The data from the FIFO is presented on this bus. It is valid only when tx_fifobyten is non-zero
tx_fifobyten[1:0]	Input	Transmit FIFO read data byte enable. The data on this bus is valid. 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 this signal does not have the value 2'b11.
tx_fifoavail	Input	Transmit FIFO data available. When asserted, this indicates that the FIFO has data ready for transmission.
tx_fifoeof	Input	Transmit FIFO end of frame. This signal is asserted along with the word/byte of frame data indicating the end of the frame.
tx_fifoempty	Input	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 underrun condition is transferred to the network through the txer signal.
tx_sndpaustim[15:0]	Input	PAUSE Frame Timer. This indicates the Pause time value that should be sent in the PAUSE frame.
tx_sndpausreq	Input	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	FIFO Control Frame. This signal indicates whether the current frame in FIFO is control frame or data frame. It is qualified by the tx_avail signal. 1 = Control frame 0 = Normal frame
tx_staten	Output	Transmit Statistics Vector Enable. When asserted, the contents of the statistics vector bus tx_statvec is valid.
tx_macread	Output	Transmit FIFO read. This is the Tri-Speed MAC Transmit FIFO read request, asserted by Tri-Speed MAC for one clock only when it intends to read the FIFO.

Table 3. Tri-Speed MAC Input and Output Signals (Continued)

Signal	Type	Description
tx_statvec[30:0]	Output	Transmit Statistics Vector. This includes useful information about the frame that was transmitted. The breakup of the bus is given below: 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 underrun 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	Transmit Done. This signal is asserted for one clock cycle after transmitting a frame if no errors in transmission.
tx_discfrm	Output	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: A FIFO under-run Late collision (10/100 Mode only) Excessive Collisions (10/100 Mode only) The user application normally moves the pointer to the next frame in these conditions.
Management Interface Signals		
mdi	Input	Management Data Input. Used to transfer information from the PHY to the management module.
mdo	Output	Management Data Output. Used to transmit information from the management module to the PHY.
mdio_en	Output	Management Data Out enable. Asserted whenever mdo is valid. Used to create a bi-directional signal.
G/MII Signals		
txd[7:0]	Output	Transmit Data. Data sent to the PHY chip.
txen	Output	Transmit Enable. Asserted by the Tri-Speed MAC to indicate that the txd bus contains valid frame.
txer	Output	Transmit Error. Asserted when the Tri-Speed MAC generates a coding error on the byte currently being transferred.
rxdv	Input	Receive Data Valid. Indicates that the data on the rxd bus is valid.
rxd[7:0]	Input	Receive Data Bus. Data is driven by the PHY on these lines and is valid whenever rxdv is asserted.
rxer	Input	Receive Data Error. This is asserted by the PHY when it detects an error.
Receive MAC Application Interface		
rx_fifo_full	Input	Rx FIFO Full. This signal indicates that the Rx FIFO is full and it cannot accept any more data. This is an error condition and should never happen.
rx_write	Output	Rx FIFO Write. This signal is driven by the Tri-Speed MAC core to request a FIFO write.
rx_dbout[15:0]	Output	Rx FIFO data Output. This bus contains the data that is to be written into the Receive FIFO.
rx_byten[1:0]	Output	Rx FIFO Byte Enable. Indicates which of the bytes in the rx_dbout bus is valid.
rx_stat_vector[31:0]	Output	Rx Statistics Vector. This bus indicates the happening of events. The definition of each signal is explained in the Transmit MAC section. This bus is qualified by the rx_stat_en signal.
rx_stat_en	Output	Rx Statistics Vector Enable. When asserted, this indicates that the contents of the rx_stat_vector bus is valid.

Table 3. Tri-Speed MAC Input and Output Signals (Continued)

Signal	Type	Description
ignore_next_pkt	Input	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	End Of Frame. Indicates that all the data for the current packet has passed on to the FIFO.
rx_error	Output	Receive Packet Error. When asserted, it indicates that the packet has an error in it. This signal is qualified with the rx_eof signal.
Rx_fifo_error	Output	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.

Related Information

For more information regarding verification and usage of the core, please refer to the User's Guide.

Appendix for ORCA[®] Series 4 FPGAs or FPSCs

Table 4. Performance and Resource Utilization¹

Name of Parameter File	Mode (Mbps)	CPU_Data_Width	MIIM_Module	PFU ²	External Pins ³	sysMEM EBRs	f _{MAX} (MHz)
ether_1gig_04_1_001.lpc	1000	16	No	451	24	1	125
ether_1gig_04_1_002.lpc	1000	8	No				
ether_1gig_04_1_003.lpc	1000	16	Yes				
ether_1gig_04_1_004.lpc	1000	8	Yes				
ether_fast_04_1_001.lpc	10/100	16	No	552	15	2	62.5
ether_fast_04_1_002.lpc	10/100	16	Yes				
ether_fast_04_1_003.lpc	10/100	8	Yes				
ether_fast_04_1_004.lpc	10/100	8	No				

1. Performance and utilization characteristics are generated using an OR4E062PBGAM680-DB in ORCA Foundry 2001 software. Synthesized using Exemplar, Version x.x(TBD). When using this IP core in a different density, package, speed, or grade within the ORCA 4 family, performance may vary slightly.
2. PFU is a standard logic block of any Lattice device. For more information, check the data sheet of the device.
3. Actual I/O usage depends on how the core is integrated within the user's design. Based on typical usage, a certain class of signals can be expected to be routed through the device's I/O resources. The numbers given here are based on only the G/MII signals being routed off the chip.

Supplied Netlist Configurations

The Ordering Part Number (OPN) for all 1000Mbps configurations of this core on ORCA Series 4 is ETHER-1GIG-04-N1. The Ordering Part Number (OPN) for all 10/100Mbps configurations of this core on ORCA Series 4 is ETHER-FAST-04-N1. Table 4 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.

To load the preset parameters for this core, click on the "Load Parameters" button inside the IP Manager tool. Make sure that you are looking for a file inside of this core's directory location. The Lattice Parameter Configuration files (.lpc) are located within the .lpc folder inside this directory.