10-Gigabit Ethernet MAC v8.0

DS201 July 13, 2006 **Product Specification**

Introduction

The LogiCORE™ 10-Gigabit Ethernet MAC core is a single-speed full-duplex 10 Gbps Ethernet Media Access Controller (MAC) solution that enables the design of high-speed Ethernet systems and subsystems.

Features

- Designed to 10-Gigabit Ethernet specification *IEEE 802.3ae-2002*
- Choice of external XGMII or internal FPGA interface to PHY layer
- Cut-through operation with minimum buffering for maximum flexibility in client-side interfacing
- Supports Deficit Idle Count for maximum data throughput; maintains minimum IFG under all conditions and provides line rate performance
- Configured and monitored through a microprocessor-neutral management interface
- Comprehensive statistics gathering with statistic vector outputs
- Supports flow-control in both directions
- MDIO STA master interface to manage PHY layers
- Extremely customizable; trade resource usage against functionality
- Available under SignOnce license program
- Delivered through the Xilinx CORE Generator™
- Supports VLAN, jumbo frames, and WAN mode
- Custom Preamble mode

LogiCORE Facts				
Core Specifics				
Device Family	Virtex™-II, Virtex-II Pro, Virtex-4, Virtex-5			
Speed Grades	• -5 for Virtex-II, Virtex-II Pro • -10 for Virtex-4 • -1 for Virtex-5			
Resources Used(1)	**Slices**	**LUTs**	**FFs**	**Block RAM**
	3616	4607	3808	0
Provided with Core				
Documentation	Product Specification User Guide Getting Started Guide			
Design File Formats	EDIF and NGC netlist			
Constraints File	UCF			
Verification	VHDL test bench Verilog test fixture			
Example Design	VHDL and Verilog			
Design Tool Requirements				
Xilinx Implementation Tools	ISE™ 8.2i			
Simulation	Mentor ModelSim® v6.1e Cadence® IUS			
Synthesis	XST			
Support				
Provided by Xilinx, Inc. @ **www.xilinx.com/support**				

1. Numbers are approximate for default configuration. See "Device Utilization" on page 18 for a complete description device utilization by configuration.

Applications

Figure 1 shows a typical Ethernet system architecture and the 10-Gigabit Ethernet MAC core within it. The MAC and all the blocks to the right are defined in Ethernet IEEE specifications.

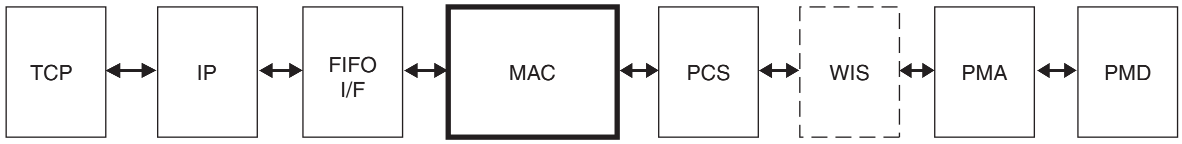

Figure 1: **Typical Ethernet System Architecture**

Figure 2 displays the 10-Gigabit Ethernet MAC core connected to a physical layer (PHY) device, for example, an optical module using the XGMII interface.

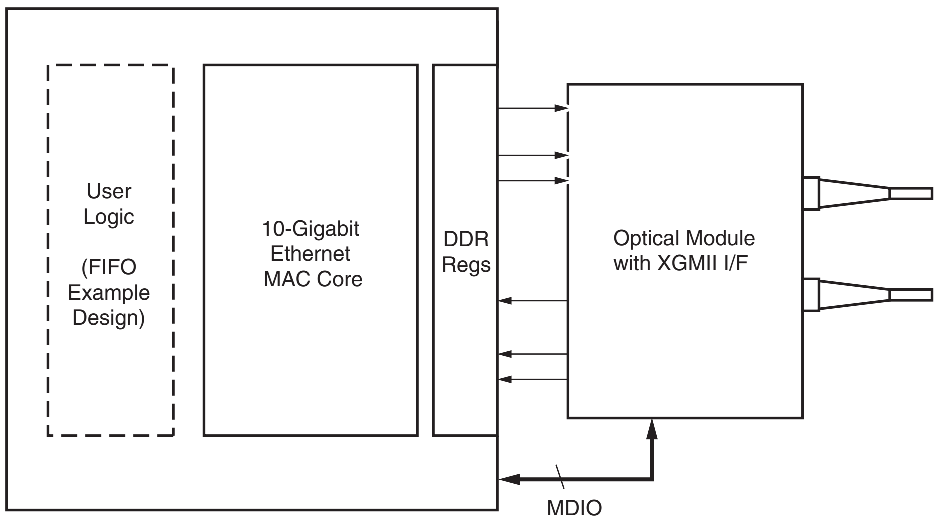

Figure 2: **10-Gigabit Ethernet MAC Core Connected to PHY with XGMII Interface**

The 10-Gigabit Ethernet MAC core is designed to be easily attached to the Xilinx XAUI core, available from http://www.xilinx.com/systemio/xaui/index.htm providing the advantage (over XGMII) of reduced pin count and improved operating distance. Figure 3 illustrates the two cores in a system using an XPAK optical module. In this case, the XGMII interface is omitted from the 10-Gigabit Ethernet MAC core at customization time and the internal FPGA fabric interface is used to interface to the XAUI core.

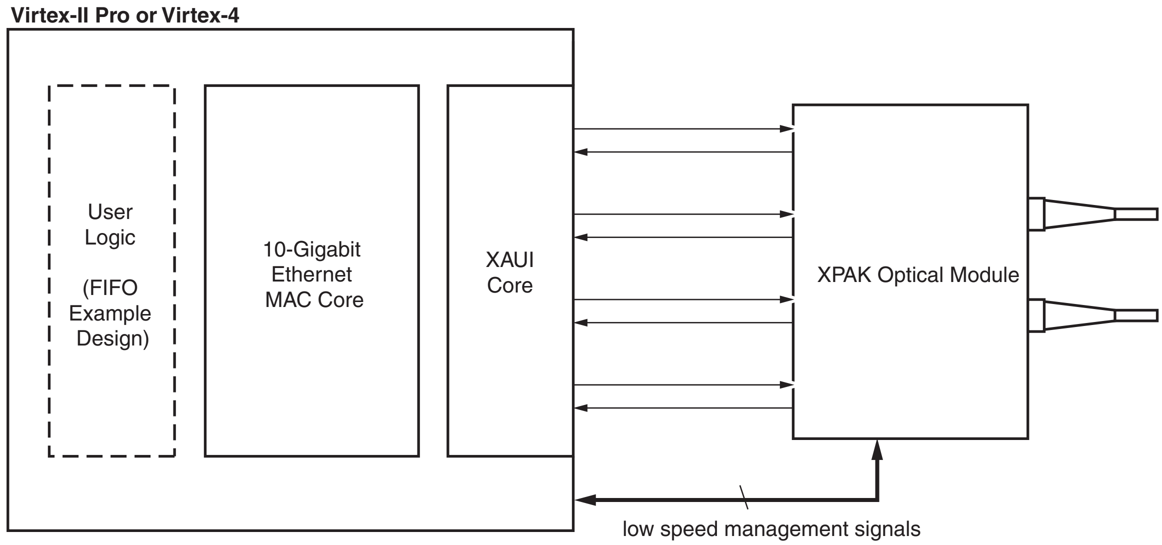

Figure 3: **10-Gigabit Ethernet MAC Core Used with Xilinx XAUI Core**

Functional Description

Figure 4 illustrates a block diagram of the 10-Gigabit Ethernet MAC core implementation. The major functional blocks of the core include the following:

- Client-side interface: Designed for simple attachment of user logic
- Transmitter
- Receiver
- Flow Control block: Implements both Receive Flow Control and Transmit Flow Control
- Reconciliation Sublayer (RS): Processes XGMII Local Fault and Remote Fault messages and handles DDR conversion
- Management interface and MDIO (optional)
- Statistics counters (optional)
- XGMII interface: Connection to the physical layer device or logic

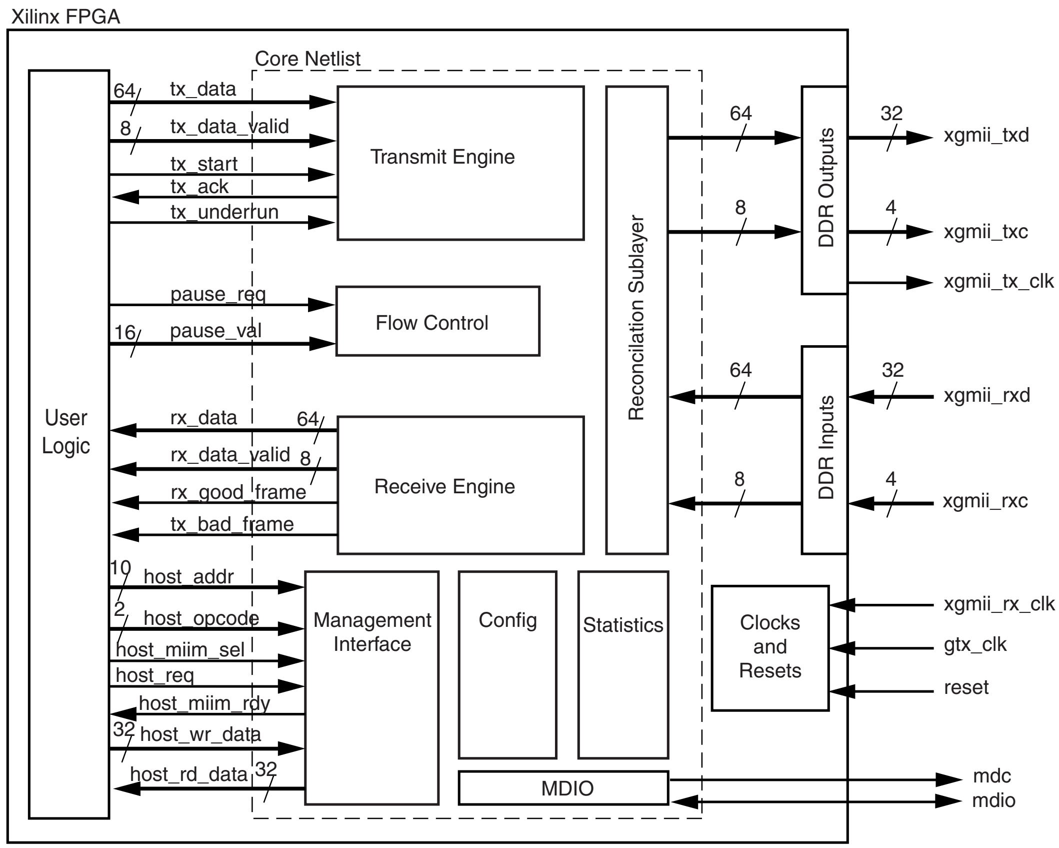

Figure 4: **Implementation of the 10-Gigabit Ethernet MAC Core**

Core Interfaces

Client-side Interface: Transmit

The client-side interface on transmit has a 64-bit data path with eight control bits to delineate bytes within the 64-bit port. Additionally, there are signals to handshake the transfer of data into the core. An example design which includes source code for a FIFO with Local Link interface is provided with the core generated by Coregen. Table 1 defines the signals.

Table 1: **Transmit Client-side Interface Port Description**

Name	Direction	Description
tx_data[63:0]	Input	Frame data to be transmitted is supplied on this port.
tx_data_valid[7:0]	Input	Control signals for tx_data port. Each asserted signal on tx_data_valid signifies which bytes of tx_data are valid; that is, if tx_data_valid[0] is ‘1,’ the signals tx_data[7:0] are valid.
tx_start	Input	Handshaking signal. Asserted by the client to make data available for transmission.

Table 1: **Transmit Client-side Interface Port Description *(Continued)***

Name	Direction	Description
tx_ack	Output	Handshaking signal. Asserted when the first column of data on tx_data has been accepted.
tx_underrun	Input	Assert this pin to forcibly corrupt the current frame.
tx_ifg_delay[7:0]	Input	Control signal for configurable inter-frame gap adjustment.

Figure 5 shows transmitting a frame through the client-side interface.

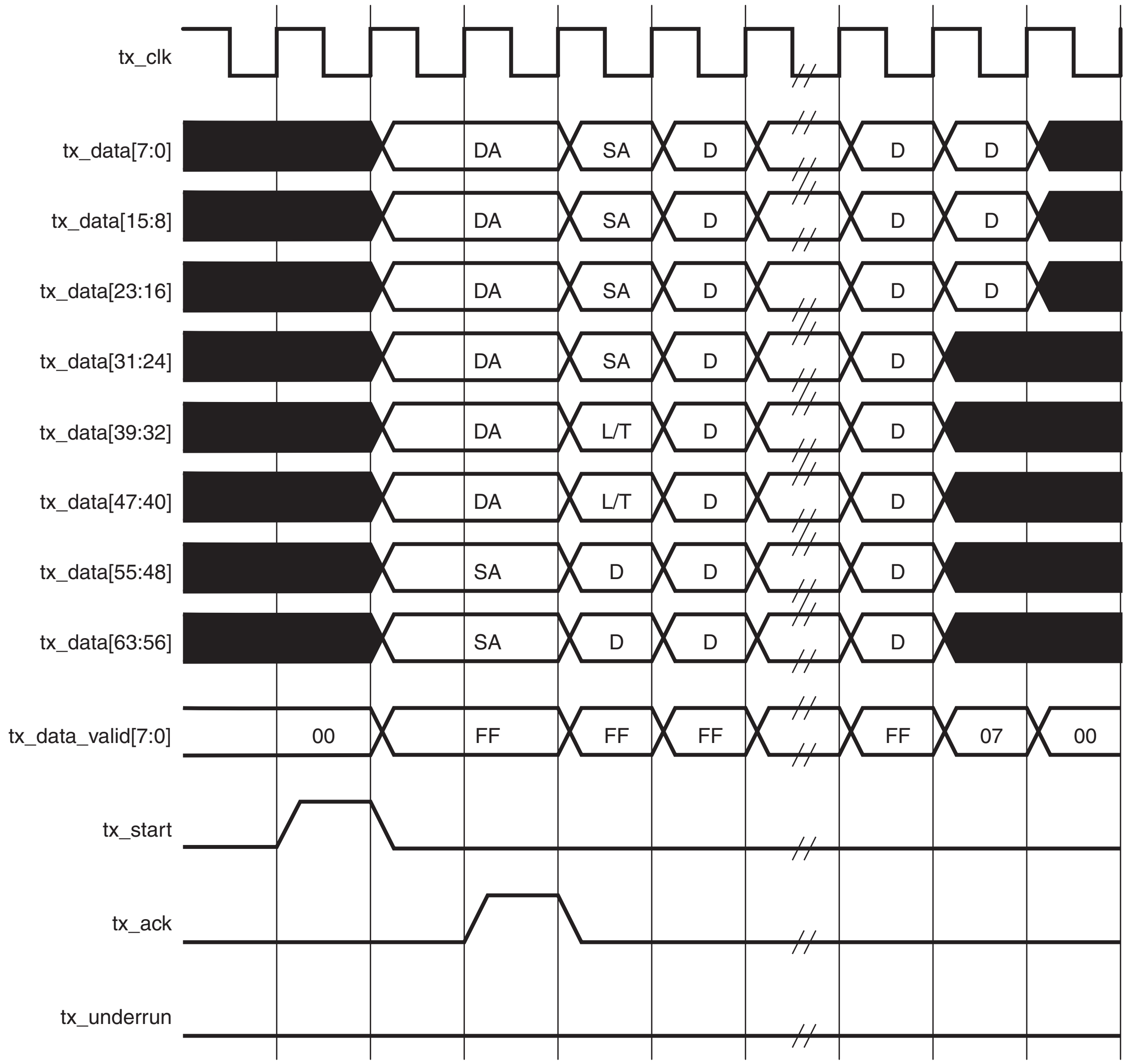

Figure 5: **Frame Transmission Across Client-side Interface**

Client-side Interface: Receive

The client-side interface on receive has a 64-bit data path with eight control bits to delineate bytes within the 64-bit port. Additionally, there are signals to indicate to the user logic the validity of the previous frame received. The ports are shown in Table 2.

Table 2: **Receive Client-side Interface Port Description**

Name	Direction	Description
rx_data[63:0]	Output	Frame data received.
rx_data_valid[7:0]	Output	Control signals for the rx_data port. Each asserted signal on rx_data_valid indicates which bytes of rx_data are valid. For example, if rx_data_valid[0] is '1,' then rx_data[7:0] is valid.
rx_good_frame	Output	Asserted at the end of frame to indicate the frame was successfully received and should be processed by the user logic.
rx_bad_frame	Output	Asserted at the end of frame to indicate the frame was not successfully received and should be discarded by the user logic.

Figure 6 shows receiving a frame through the client-side interface.

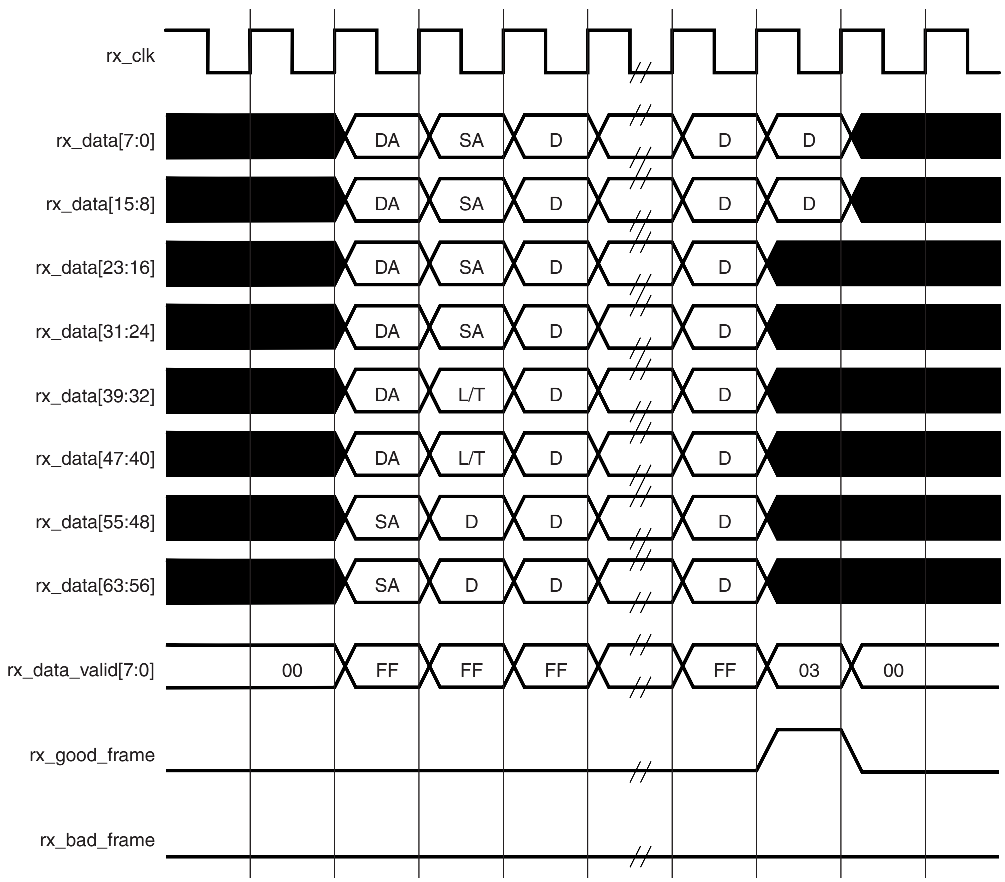

Figure 6: **Frame Reception Across Client-side Interface**

Flow Control Interface

The flow control interface is used to initiate the transmission of flow control frames from the core. The ports associated with this interface are shown in Table 3.

Table 3: **Flow Control Interface Port Description**

Name	Direction	Description
pause_req	Input	Request that a flow control frame is emitted from the MAC core.
pause_val[15:0]	Input	Pause value field for flow control frame to be sent when pause_req asserted.

Figure 7 illustrates a flow control request. Asserting the pause_req signal in this way results in a flow control frame being transmitted from the core with the PAUSE VALUE field set to pause_val[15:0]

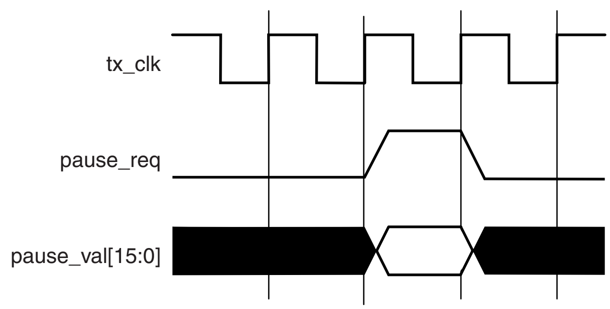

Figure 7: **Flow Control Frame Request**

XGMII Interface or 64-bit SDR PHY Interface

The PHY interface may be a 32-bit DDR XGMII interface a or 64-bit SDR interface, depending on the customization of the core. This interface is used to connect to the physical layer, whether this is a separate device or implemented in the FPGA beside the MAC core. Tables 4 and 5 show the ports associated with this interface.

Table 4: **32-bit XGMII PHY Interface Port Descriptions**

Name	Direction	Description
xgmii_tx_clk	Output	Forwarded clock to PHY device.
xgmii_txd[31:0]	Output	Transmit data to PHY; double data rate (DDR) source centred on xgmii_tx_clk.
xgmii_txc[3:0]	Output	Transmit control to PHY; DDR source-centred on xgmii_tx_clk.
xgmii_rx_clk	Input	Inbound clock from PHY device.
xgmii_rxd[31:0]	Input	Received data from PHY; DDR source-centred on xgmii_rx_clk.
xgmii_rxc[3:0]	Input	Received control from PHY; DDR source-centred on xgmii_rx_clk.

Table 5: **64-bit SDR PHY Interface Port Descriptions**

Name	Direction	Description
xgmii_txd[63:0]	Output	Transmit data to PHY. Synchronous to rising edge of tx_clk.
xgmii_txc[7:0]	Output	Transmit control to PHY. Synchronous to rising edge of tx_clk.
xgmii_rx_clk	Input	Inbound clock from PHY.
xgmii_rxd[63:0]	Input	Received data from PHY. Synchronous to rising edge of xgmii_rx_clk.
xgmii_rxc[7:0]	Input	Received control from PHY. Synchronous to rising edge of xgmii_rx_clk.

Management Interface

Configuration of the core, access to the statistics block, and access to the MDIO port can be provided through the management interface, a 32-bit processor-neutral interface independent of the Ethernet data path. Table 6 define the ports associated with the management interface.

Table 6: **Management Interface Port Descriptions**

Name	Direction	Description
host_clk	Input	Clock for management interface. Range between 10 MHz and 133 MHz.
host_opcode[1:0]	Input	Defines operation to be performed over management interface.
host_addr[9:0]	Input	Address of register to be accessed.
host_wr_data[31:0]	Input	Data to write to register.
host_rd_data[31:0]	Output	Data read from register.
host_miim_sel	Input	When asserted, the MDIO interface is accessed.
host_req	Input	Used to request a transaction on the MDIO interface or read from the statistic registers.
host_miim_rdy	Output	When asserted, the MDIO interface has completed any pending transaction and is ready for a new transaction.

The management interface can be omitted at core customization; if omitted, Configuration Signals are available instead.

Configuration Registers

After the core is powered up and reset, the user application can reconfigure some of the core parameters from their default values, such as flow control operation and WAN mode. Configuration registers can be written and read at any time; however, changes may only take effect during the next inter-frame gap period. Exceptions to this include the soft reset registers which take effect immediately.

Configuration of the 10-Gigabit Ethernet MAC core is performed through a bank of registers accessed through the management interface. The configuration registers in this bank and their addresses are shown Table 7.

Table 7: **Configuration Registers**

Address (Hex)	Description
0x200	Receiver Configuration Word 0.
0x240	Receiver Configuration Word 1.
0x280	Transmitter Configuration.
0x2C0	Flow Control Configuration.
0x300	Reconciliation Sublayer Configuration.
0x340	Management Configuration.

The contents of each configuration register are defined in Tables 8 through 13.

Table 8: **Receiver Configuration Word 0**

Bit	Default Value	Description
31:0	All 0s	Pause frame MAC address [31:0].

Table 9: **Receiver Configuration Word 1**

Bit	Default Value	Description
15:0	All 0s	Pause frame MAC address [47:32].
25:16	N/A	Reserved.
26	0	Receiver Preserve Preamble Enable.
27	0	VLAN Enable.
28	1	Receiver Enable.
29	0	In-band FCS Enable.
30	0	Jumbo Frame Enable.
31	0	Receiver Reset.

Table 10: **Transmitter Configuration Word**

Bit	Default Value	Description
22:0	N/A	Reserved.
23	0	Transmitter Preserve Preamble Enable.
24	0	Deficit Idle Count Enable.
25	0	Inter-Frame Gap Adjust Enable.
26	0	WAN Mode Enable.
27	0	VLAN Enable.

Table 10: **Transmitter Configuration Word *(Continued)***

Bit	Default Value	Description
28	1	Transmitter Enable.
29	0	In-band FCS Enable.
30	0	Jumbo Frame Enable.
31	0	Transmitter Reset.

Table 11: **Flow Control Configuration Word**

Bit	Default Value	Description
28:0	N/A	Reserved.
29	1	Receive Flow Control Enable.
30	1	Transmit Flow Control Enable.
31	N/A	Reserved.

Table 12: **Reconciliation Sublayer Configuration Word**

Bit	Default Value	Description
26:0	N/A	Reserved.
27	0	Fault Inhibit.
28	N/A	Local Fault received. Read-only.
29	N/A	Remote Fault received. Read-only.
30	N/A	Transmit DCM Locked. Read-only.
31	N/A	Receive DCM Locked. Read-only.

Table 13: **Management Configuration Word**

Bit	Default Value	Description
4:0	All 0s	Clock Divide[4:0]. Used as a divider value to generate MDC signal at 2.5 MHz.
5	0	MDIO Enable.
31:6	N/A	Reserved.

When accessing the configuration registers through the management interface, the upper bit of host_opcode functions as an active-low write-enable signal. Figure 8 illustrates a configuration register read and Figure 9 illustrates a configuration register write. The lower bit of the host_opcode is only used for MDIO accesses

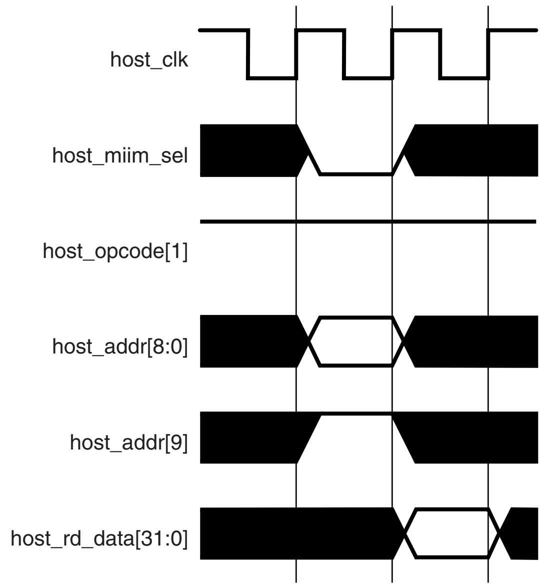

Figure 8: **Configuration Register Read Timing**

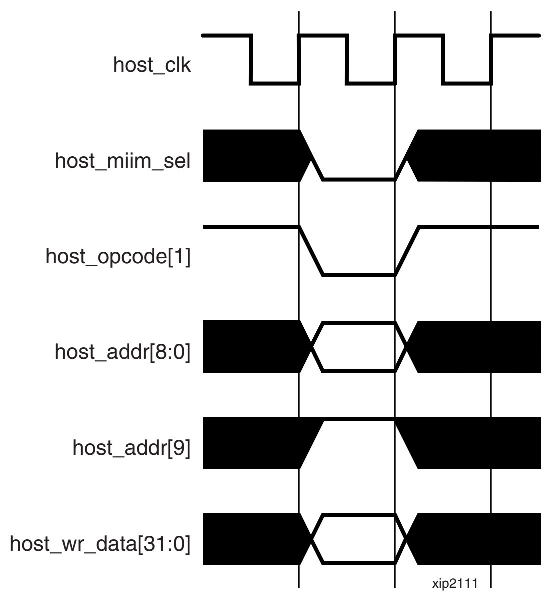

Figure 9: **Configuration Register Write Timing**

Statistic Counters

During operation, the MAC core collects statistics on the success and failure of various operations, for processing by network management entities elsewhere in the system. These statistics are accessed through the management interface. A list of statistics is shown in Table 14.

Table 14: **Statistic Counters**

Address (hex)	Name
0x000	Frames Received OK.
0x001	Frame Check Sequence Errors.
0x002	Broadcast frames Received OK.
0x003	Multicast Frames Received OK.
0x004	64 byte Frames Received OK.
0x005	65-127 byte Frames Received OK.
0x006	128-255 byte Frames Received OK.
0x007	256-511 byte Frames Received OK.
0x008	512-1023 byte Frames Received OK.
0x009	1024-MaxFrameSize byte Frames Received OK.
0x00A	Control Frames Received OK.
0x00B	Length/Type Out of Range.
0x00C	VLAN Tagged Frames Received OK.
0x00D	Pause Frames Received OK.
0x00E	Control Frames Received with Unsupported Opcode.
0x00F	Oversize Frames Received OK.
0x010	Undersized Frames Received.
0x011	Fragment Frames Received.
0x012	Number of Bytes Received.
0x013	Number of Bytes Transmitted.
0x020	Frames Transmitted.
0x021	Broadcast Frames Transmitted.
0x022	Multicast Frames Transmitted.
0x023	Underrun Errors.
0x024	Control Frames Transmitted OK.
0x025	64 byte Frames Transmitted OK.
0x026	65-127 byte Frames Transmitted OK.
0x027	128-255 byte Frames Transmitted OK.
0x028	256-511 byte Frames Transmitted OK.
0x029	512-1023 byte Frames Transmitted OK.

Table 14: **Statistic Counters *(Continued)***

Address (hex)	Name
0x02A	1024-MaxFrameSize byte Frames Transmitted OK.
0x02B	VLAN Tagged Frames Transmitted OK.
0x02C	Pause Frames Transmitted OK.
0x02D	Oversize Frames Transmitted OK.

Figure 10 shows a statistics register access across the management interface. Each register is 64-bits wide and for this reason must be read in a two-cycle transfer.

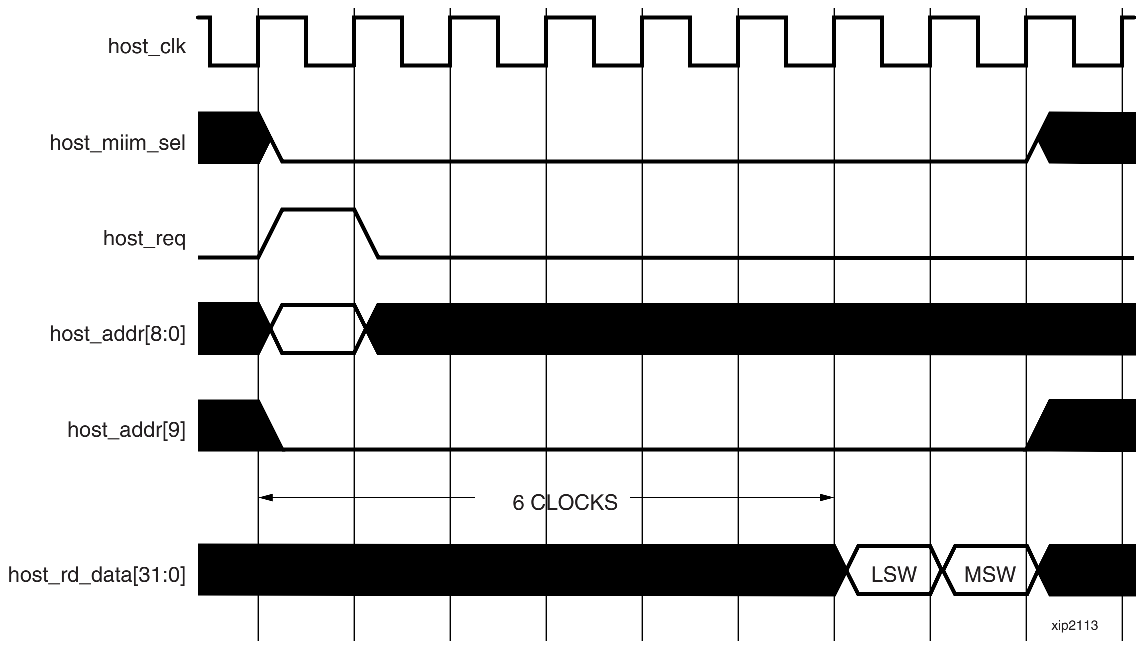

Figure 10: **Statistics Register Read Across Management Interface**

The statistic counters are an optional block of the 10-Gigabit Ethernet MAC core.

MDIO STA Master

The MDIO STA master interface implemented in the 10-Gigabit Ethernet MAC core is an STA entity (as defined by *IEEE Std.802.3ae-2002*) that can initiate transactions to one or more attached physical layer MDIO Managed Devices (MMDs). Table 15 defines the ports associated with this interface.

Table 15: **MDIO Port Descriptions**

Name	Direction	Description
mdc	Output	Management Clock. Derived from host_clk.
mdio_in	Input	Serial data in.
mdio_out	Output	Serial data out.
mdio_tri	Output	Tristate control for MDIO signals; ‘0’ indicates that mdio_out should be asserted onto the MDIO bus.

Figure 11 shows an MDIO transaction across the management interface For MDIO transactions, host_opcode maps into the OP field of the MDIO frame, host_addr maps to the two address fields of the frame (PRTAD and DEVAD), and host_wr_data[15:0] maps into the address/data field of the MDIO frame for Address or Write transactions, and the address/data field maps into host_rd_data[15:0] for Read or Read/Increment transactions.

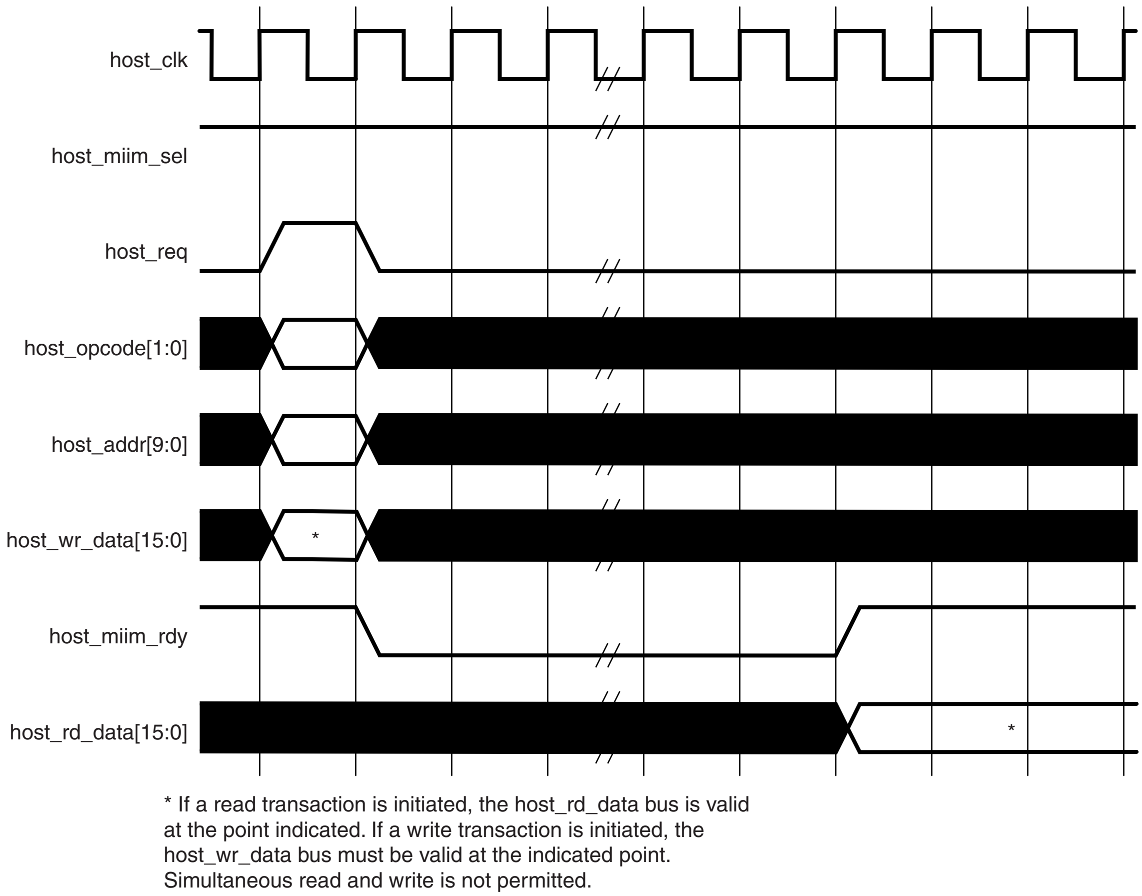

Figure 11: **MDIO Transaction Across Management Interface**

If the management interface is omitted from the core, the MDIO interface is also omitted.

Configuration Signals

If the management interface is omitted at core customization time, a configuration vector is exposed by the core. This allows the user to configure the core by statically or dynamically driving the constituent bits of the port. Table 16 describes the configuration signal. For more information about the use of the configuration vector, see the *10-Gigabit Ethernet MAC User Guide*.

Table 16: **Configuration Signal**

Name	Direction	Description
configuration_vector[66:0]	Input	Configuration signals for the core. The bits are: Bits 47 to 0: Pause frame MAC Source Address Bit 48: Receive VLAN Enable Bit 49: Receive Enable Bit 50: Receive In-Band FCS Bit 51: Receive Jumbo Frame Enable Bit 52: Receiver Reset Bit 53: Transmitter WAN Mode Bit 54: Transmitter IFG Adjust Bit 55: Transmitter VLAN Enable Bit 56: Transmitter Enable Bit 57: Transmitter In-Band FCS Enable Bit 58: Transmitter Jumbo Frame Enable Bit 59: Transmitter Reset Bit 60: Receive Flow Control Enable Bit 61: Transmit Flow Control Enable Bit 62: Deficit Idle Count Enable Bit 63: Reserved — Tie to ‘0’ Bit 64: RS Fault Inhibit Bit 65: Transmitter Preserve Preamble Enable Bit 66: Receiver Preserve Preamble Enable

Statistic Vectors

In addition to the statistic counters described in Management Interface, there are two statistics vector outputs on the core netlist that are used to signal the core state. These vectors are actually used as the inputs of the counter logic internal to the core, so if, for example, a user omits the statistic counters at the CORE Generator customization stage, a relevant subset can be implemented in user logic. Table 17 identifies the signals. The contents of the vectors are defined in Tables 18 and 19.

Table 17: **Statistic Vector Signals**

Name	Direction	Description
tx_statistics_vector[24:0]	Output	Aggregated statistics flags for transmitted frame.
tx_statistics_valid	Output	Valid strobe for tx_statistics_vector.
rx_statistics_vector[28:0]	Output	Aggregated statistics flags for received frames.
rx_statistics_valid	Output	Valid strobe for rx_statistics_vector.

Table 18: **Transmit Statistics Vector Contents**

tx_statistics_vector bits	Name
24	PAUSE_FRAME_TRANSMITTED
23 to 20	BYTES_VALID
19	VLAN_FRAME
18 to 5	FRAME_LENGTH_COUNT
4	CONTROL_FRAME
3	UNDERRUN_FRAME
2	MULTICAST_FRAME
1	BROADCAST_FRAME
0	SUCCESSFUL_FRAME

Table 19: **Receive Statistics Vector Contents**

rx_statistics_vector bits	Name
28	LEN_TYPE_RANGE
27	BAD_OPCODE
26	FLOW_CONTROL_FRAME
25 to 22	BYTES_VALID
21	VLAN_FRAME
20	OUT_OF_BOUNDS
19	CONTROL_FRAME
18 to 5	FRAME_LENGTH_COUNT
4	MULTICAST_FRAME
3	BROADCAST_FRAME
2	FCS_ERROR
1	BAD_FRAME
0	GOOD_FRAME

Clocks and Resets

Table 20 describes the clock and reset ports present on the supplied example design. In the source code of the example design, other system clocks are derived from the `gtx_clk` and `xgmii_rx_clk` signals for use in the core logic. This clock arrangement may be customized in the user application as required.

Table 20: **Clock and Reset Ports**

Name	Direction	Description
gtx_clk	Input	Global transmit clock. All other transmit clocks are derived from this clock.
xgmii_rx_clk	Input	XGMII Receive clock. All receive clocks are derived from this clock.
reset	Input	Asynchronous reset

Verification

The 10-Gigabit Ethernet MAC core has been verified in both simulation and hardware testing.

Simulation

A highly parameterizable transaction-based simulation test suite has been used to verify the core. Tests include:

- Configuration register access through management interface
- Local Fault and Remote Fault handling
- Frame transmission
- Frame reception
- CRC validity
- Handling of CRC errors
- Statistic counter access through management interface and validity of counts
- Statistic vector validity
- Initiating MDIO transactions through management interface
- Use of custom preamble field

Hardware Verification

The core has been used in a number of hardware test platforms at Xilinx, including the following:

- The core has been used in a test platform design with the Xilinx 10-Gigabit Ethernet XAUI LogiCORE. This design comprises the MAC, XAUI, a *ping* loopback FIFO and a test pattern generator all under embedded PowerPC™ processor control.
- This design has been used for conformance and interoperability testing at the University of New Hampshire Interoperability Lab.

Device Utilization

Table 21 provides approximate slice counts for various customization options.

Table 21: **Device Utilization for the 10-Gigabit Ethernet MAC Core**

Parameter Values				Resource Usage		
Physical Interface	**Management Interface**	**Statistic Counters**	**Simplex Split**	**Slices**	**LUTs**	**FFs**
XGMII	Yes	Yes	None	3732	4598	4028
		No	None	2758	3347	3024
	No	No	None	2617	3176	2851
			Transmit-only	1378	1878	1474
			Receive-only	1092	1065	1191
No XGMII	Yes	Yes	None	3616	4607	3808
		No	None	2634	3347	2804
	No	No	None	2488	3176	2631
			Transmit-only	1341	1878	1402
			Receive-only	999	1065	1043

References

[1] IEEE Standard 802.3-2002, "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications."

[2] IEEE Std. 802.3ae-2002, "Media Access Control (MAC) Parameters, Physical Layers, and Management Parameters for 10 Gb/s Operation."

Support

Visit http://www.xilinx.com/support/ for technical support. Xilinx provides technical support for this LogiCORE product when used as described in product documentation.

Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not listed in the documentation or if customized beyond that allowed in the product documentation, or if any changes are made in sections of the design marked as *DO NOT MODIFY*.

Ordering Information

This LogiCORE product is provided under the SignOnce IP Site License and can be generated using Xilinx CORE Generator v8.2i or higher. The Xilinx CORE Generator is shipped with the ISE Foundation software at no additional charge.

A Simulation Only Evaluation license is included with the core using the CORE Generator. To access the full functionality of the core, including FPGA bitstream generation and testing in hardware, a full license must be obtained. For more information, please visit the 10-Gigabit Ethernet MAC page at http://www.xilinx.com/systemio/10gmac/index.htm.

Please contact your local Xilinx sales representative for pricing and availability of Xilinx LogiCORE modules and software. Information on additional Xilinx LogiCORE modules is available on the Xilinx IP Center.

Revision History

Date	Version	Revision
09/30/04	1.0	Initial Xilinx release.
4/28/05	2.0	Updated for Xilinx tools 7.1i SP1.
01/11/06	3.0	Updated for Xilinx tools 8.1i.
7/13/06	4.0	Updated to core version 8.0; Xilinx tools 8.2i.