

Introduction

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

Features

- Designed to 10-Gigabit Ethernet specification *IEEE 802.3-2008*
- Choice of external XGMII or internal FPGA interface to PHY layer (internal interface only on Spartan®-6 devices)
- 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™ software
- Supports VLAN, jumbo frames, and WAN mode (WAN mode not supported on Spartan-6® devices.)
- Custom Preamble mode

LogiCORE IP Facts				
Core Specifics				
Device Family ¹	Speed Grades			
Virtex-6	-1			
Virtex-5	-1			
Virtex-4	-10			
Spartan-6 ²	-3			
Resources Used ³	Slices	LUTs	FFs	Block RAM
	1803	3738	3652	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® software v12.1			
Simulation	Mentor Graphics ModelSim v6.5c and above Cadence Incisive Enterprise Simulator (IES) v9.2 and above Synopsys VCS and VCS MX 2009.12 and above			
Synthesis	XST 12.1			
Support				
Provided by Xilinx, Inc. @ www.xilinx.com/support				

1. For the complete list of supported devices, see the 12.1 release notes for this core.
2. External XGMII interface and WAN mode not supported on Spartan-6 devices.
3. Numbers are approximate for default configuration in Virtex-5 devices. See [Table 21](#) through [Table 23](#) for a complete description of device utilization by configuration and device family.

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.

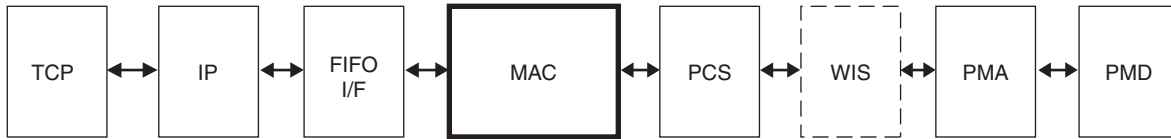


Figure 1: Typical Ethernet System Architecture

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

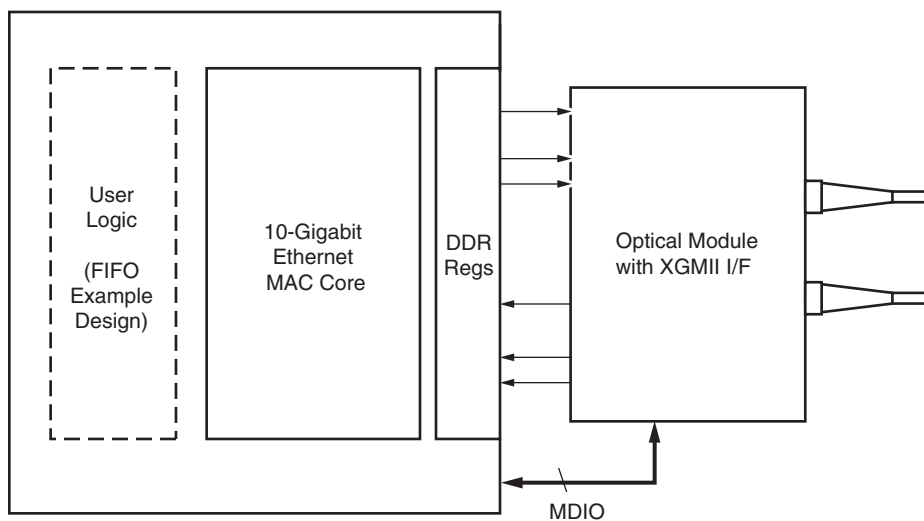


Figure 2: 10-Gigabit Ethernet MAC Core Connected to PHY with XGMII Interface

The 10-Gigabit Ethernet MAC core is designed to be easily attached to the [Xilinx IP XAUI core](#), which provides 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.

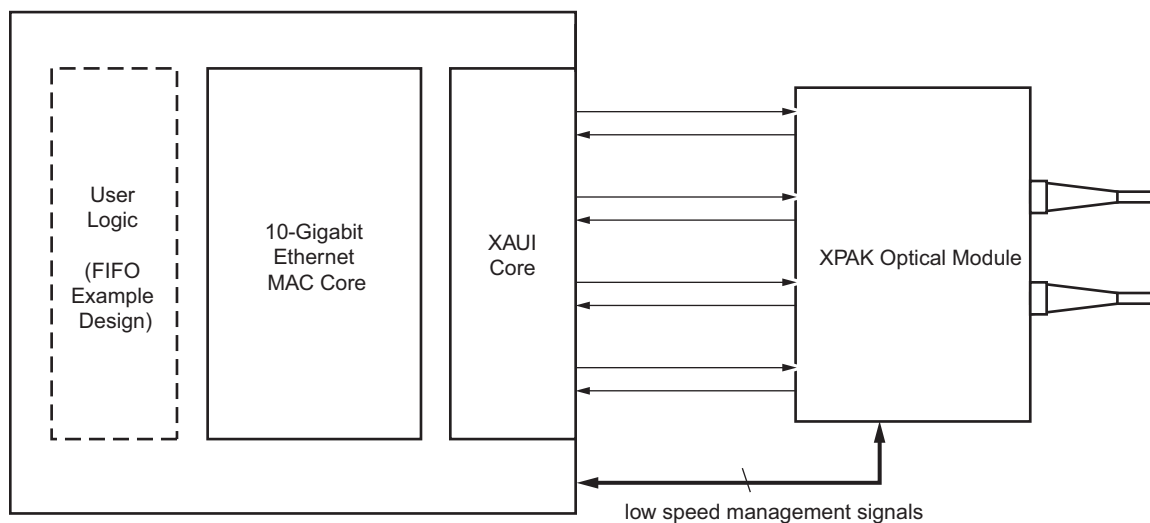


Figure 3: 10-Gigabit Ethernet MAC Core Used with Xilinx XAUI Core

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.

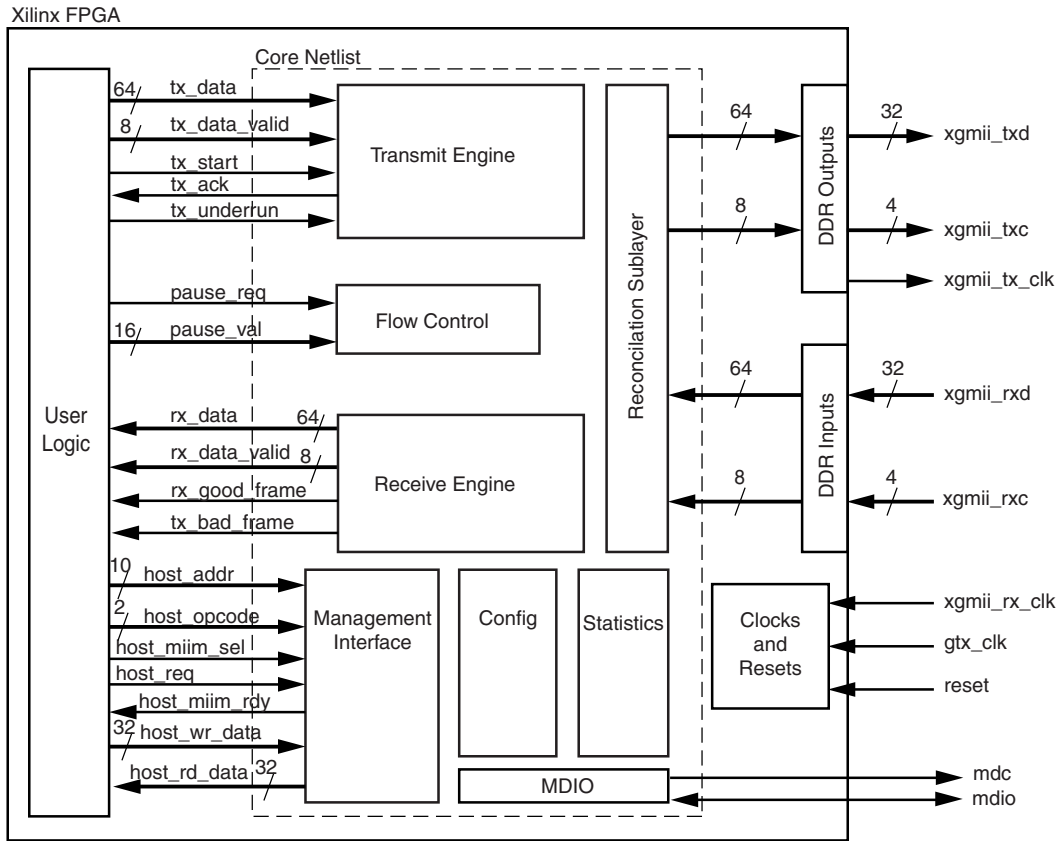


Figure 4: Implementation of the 10-Gigabit Ethernet MAC Core

Core Interfaces

Client-Side Interface: Transmit

The client-side interface on transmit has a 64-bit datapath 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 LocalLink interface is provided with the core generated by CORE Generator software. [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.
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 interframe gap adjustment.

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

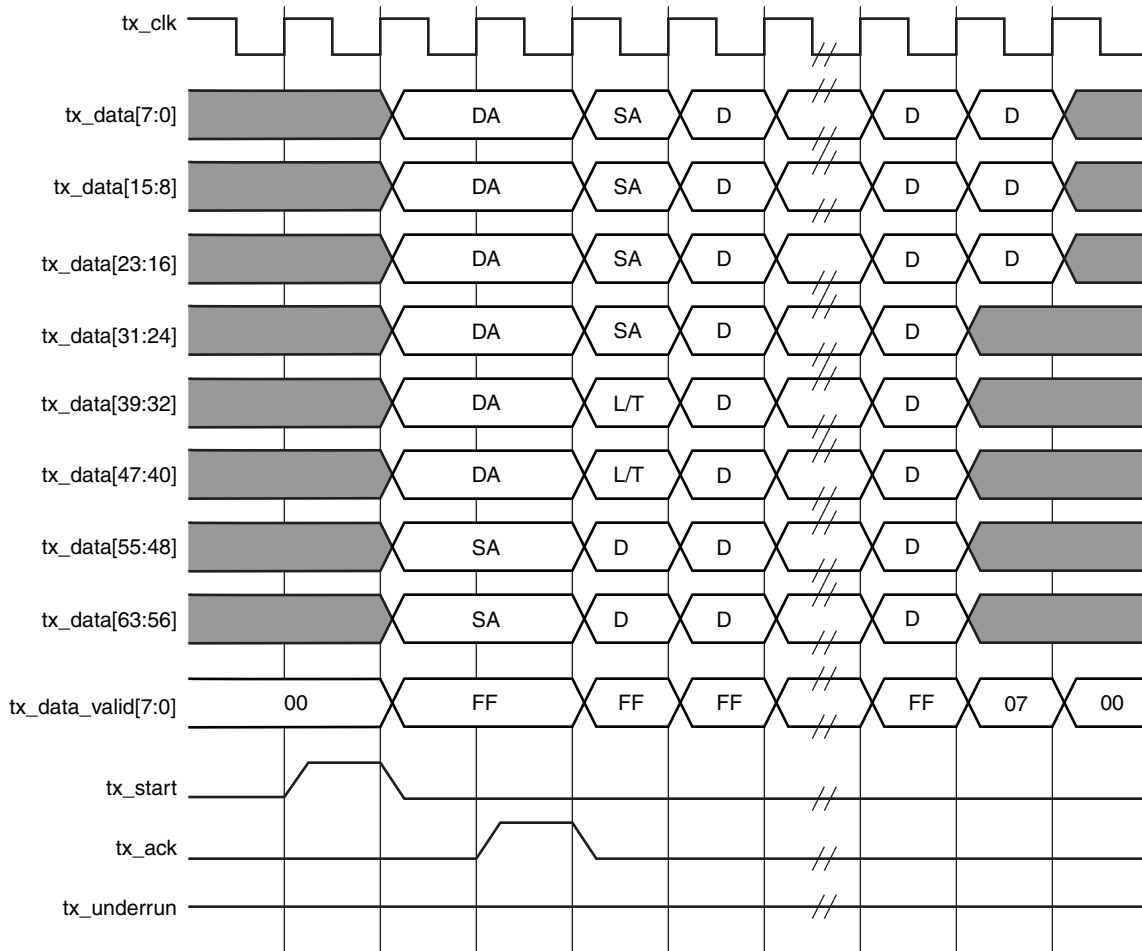


Figure 5: Frame Transmission Across Client-Side Interface

Client-Side Interface: Receive

The client-side interface on receive has a 64-bit datapath 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.

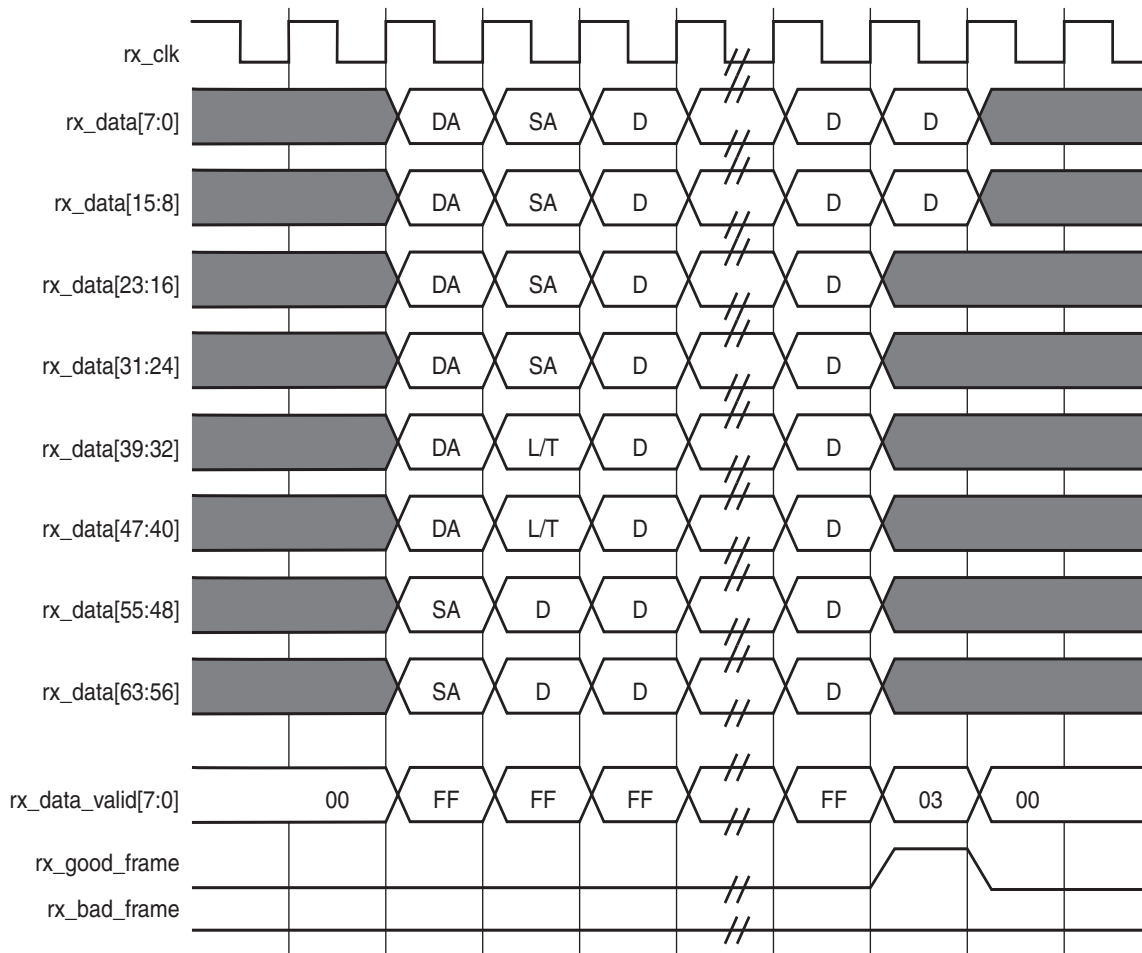


Figure 6: Frame Reception Across Client-Side Interface

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]

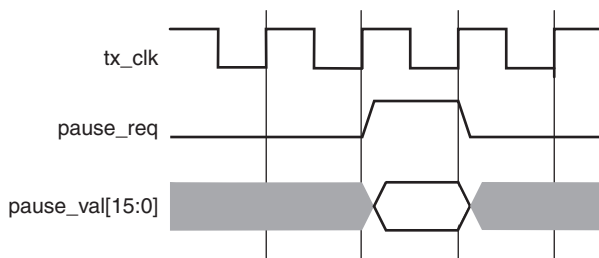


Figure 7: Flow Control Frame Request

XGMII Interface or 64-bit SDR PHY Interface

The PHY interface may be a 32-bit DDR XGMII interface or a 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. Table 4 and Table 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 datapath. [Table 6](#) defines 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. In this case, the available configuration signals would be used. See [Configuration Signals, page 15](#) for more information.

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 interframe 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 [Table 8](#) through [Table 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]
24:16	N/A	Reserved
25	0	Receiver Length/Type Check Disable
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
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.

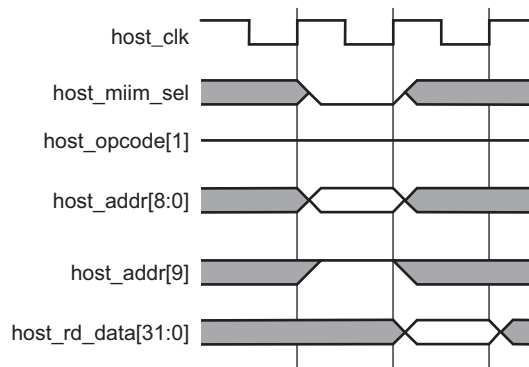


Figure 8: Configuration Register Read Timing

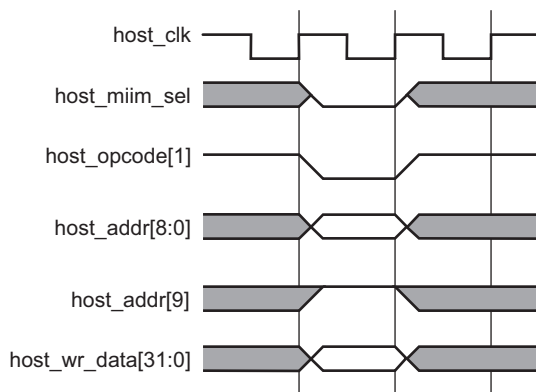


Figure 9: Configuration Register Write Timing

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

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

Address (hex)	Name
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
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.

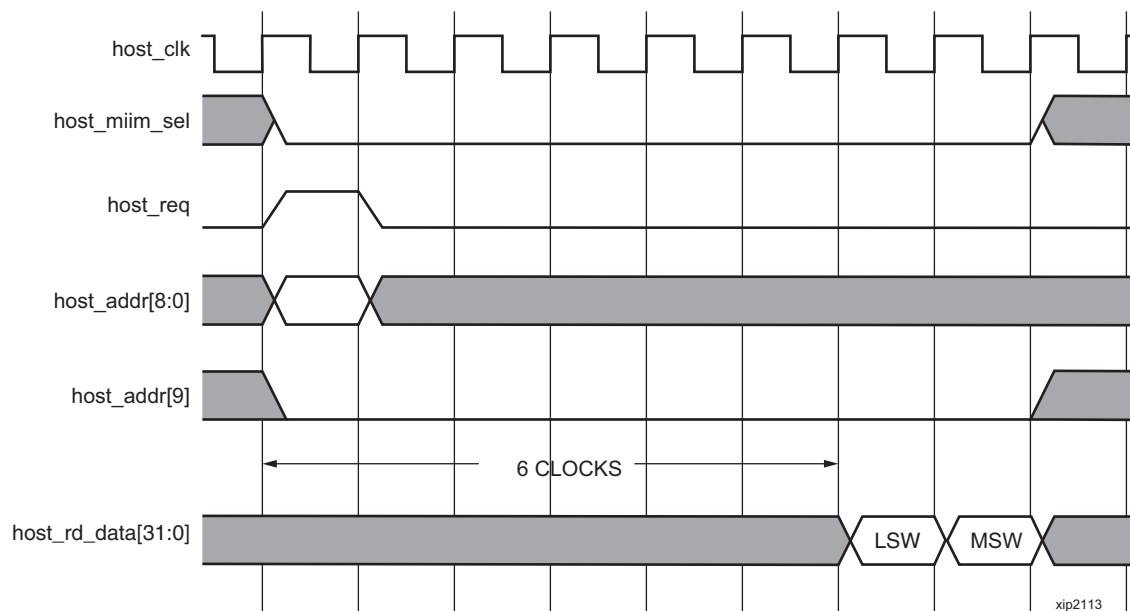


Figure 10: **Statistics Register Read Across Management Interface**

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

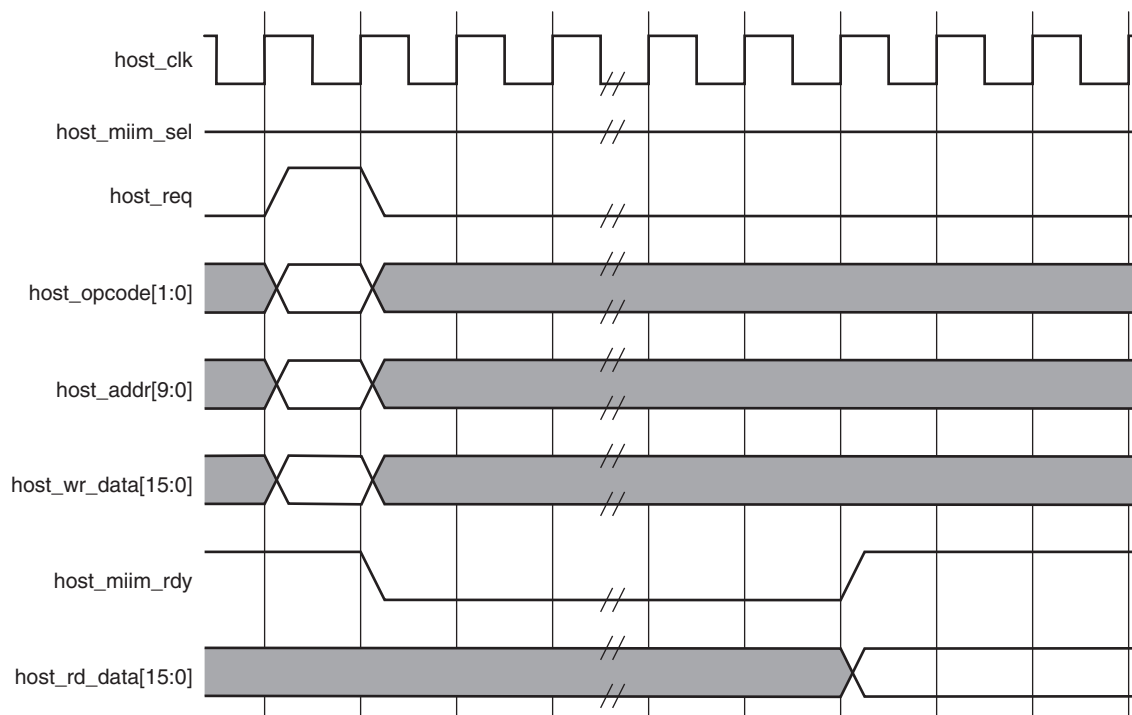
MDIO STA Master

The MDIO STA master interface implemented in the 10-Gigabit Ethernet MAC core is an STA entity (as defined by *IEEE Std.802.3-2008* 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 <code>host_clk</code>
mdio_in	Input	Serial data in
mdio_out	Output	Serial data out
mdio_tri	Output	3-state control for MDIO signals; '0' indicates that <code>mdio_out</code> 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.



* If a read transaction is initiated, the `host_rd_data` bus is valid at the point indicated. If a write transaction is initiated, the `host_wr_data` bus must be valid at the indicated point. Simultaneous read and write is not permitted.

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 you 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[68:0]	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 Bit 67: Receiver Length/Type Error Disable Bit 68: Control Frame Length Check Disable

Statistic Vectors

In addition to the statistic counters described in [Management Interface, page 9](#), 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 you omit the statistic counters at the CORE Generator software customization stage, a relevant subset can be implemented in user logic. [Table 17](#) identifies the signals. The contents of the vectors are defined in [Table 18](#) and [Table 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

Table 19: Receive Statistics Vector Contents (Continued)

rx_statistics_vector bits	Name
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
<code>gtx_clk</code>	Input	Global transmit clock; all other transmit clocks are derived from this clock
<code>xgmii_rx_clk</code>	Input	XGMII receive clock; all receive clocks are derived from this clock
<code>reset</code>	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 included:

- 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 IP. 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

The Virtex®-4 device family contains four-input LUTs; all other families contain six-input LUTs. For this reason, the device utilization for Virtex-4 FPGAs is listed separately. See either:

- [Virtex-6 FPGAs](#)
- [Virtex-5 FPGAs](#)
- [Virtex-4 FPGAs](#)
- [Spartan-6 FPGAs](#)

Virtex-6 FPGAs

Table 21 provides approximate utilization figures for various core options when a single instance of the core is instantiated in a Virtex-6 device.

Utilization figures are obtained by implementing the block-level wrapper for the core. This wrapper is part of the example design and connects the core to the selected physical interface.

Table 21: Device Utilization for the 10-Gigabit Ethernet MAC Core (Virtex-6 FPGAs)

Parameter Values				Resource Usage			
Physical Interface	Management Interface	Statistic Counters	Simplex Split	Slices	LUTs	FFs	BUFGs
XGMII	Yes	Yes	None	1543	3020	3650	2
		No	None	1222	2338	2708	2
	No	No	None	1168	2245	2468	2
			Transmit-only	653	1313	1218	0
			Receive-only	389	781	1044	2
No XGMII	Yes	Yes	None	1606	3020	3650	2
		No	None	1208	2338	2708	2
	No	No	None	1168	2245	2468	2
			Transmit-only	667	1313	1218	0
			Receive-only	398	781	1044	2

Virtex-5 FPGAs

Table 22 provides approximate utilization figures for various core options when a single instance of the core is instantiated in a Virtex-5 device.

Utilization figures are obtained by implementing the block-level wrapper for the core. This wrapper is part of the example design and connects the core to the selected physical interface.

Table 22: Device Utilization for the 10-Gigabit Ethernet MAC Core (Virtex-5 FPGAs)

Parameter Values				Resource Usage			
Physical Interface	Management Interface	Statistic Counters	Simplex Split	Slices	LUTs	FFs	BUFGs
XGMII	Yes	Yes	None	2162	3420	3647	1
		No	None	1715	2554	2712	1
	No	No	None	1596	2427	2470	1
			Transmit-only	863	1402	1221	0
			Receive-only	510	843	1044	1
No XGMII	Yes	Yes	None	2272	3420	3647	1
		No	None	1812	2554	2712	1
	No	No	None	1552	2427	2470	1
			Transmit-only	866	1402	1221	0
			Receive-only	560	843	1044	1

Virtex-4 FPGAs

Table 23 provides approximate utilization values for various core options when a single instance of the core is instantiated in a Virtex-4 device.

Utilization figures are obtained by implementing the block-level wrapper for the core. This wrapper is part of the example design and connects the core to the selected physical interface.

Table 23: Device Utilization for the 10-Gigabit Ethernet MAC Core (Virtex-4 FPGAs)

Parameter Values				Resource Usage			
Physical Interface	Management Interface	Statistic Counters	Simplex Split	Slices	LUTs	FFs	BUFGs
XGMII	Yes	Yes	None	4862	4457	3654	1
		No	None	3668	3372	2716	1
	No	No	None	3548	3194	2474	1
			Transmit-only	1843	1927	1226	0
			Receive-only	1371	1147	1045	1
No XGMII	Yes	Yes	None	4715	4457	3654	1
		No	None	3633	3372	2716	1
	No	No	None	3267	3194	2474	1
			Transmit-only	1869	1927	1226	0
			Receive-only	1371	1147	1045	1

Spartan-6 FPGAs

Table 24 provides approximate utilization figures for various core options when a single instance of the core is instantiated in a Spartan-6 device.

Utilization figures are obtained by implementing the block-level wrapper for the core. This wrapper is part of the example design and connects the core to the selected physical interface.

Table 24: Device Utilization for the 10-Gigabit Ethernet MAC Core (Spartan-6 FPGAs)

Parameter Values				Resource Usage			
Physical Interface	Management Interface	Statistic Counters	Simplex Split	Slices	LUTs	FFs	BUFGs
No XGMII	Yes	Yes	None	1696	2888	3742	1
		No	None	1264	2206	2804	1
	No	No	None	1200	2107	2565	1
			Transmit-Only	709	1225	1241	0
			Receive-Only	439	737	1116	1

References

[1] IEEE Standard 802.3-2008, "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications."

Support

Visit www.xilinx.com/support/ for technical support. Xilinx provides technical support for this LogiCORE IP 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 Xilinx LogiCORE IP module is provided under the SignOnce IP Site License. Two free evaluation licenses are provided: The Simulation Only license is provided with the CORE Generator software, and the Full System Hardware Evaluation license, which lets you test your designs in hardware for a limited period of time, can be downloaded from the [10GEMAC product page](#).

For full access to all core functionality, both in simulation and in hardware, you must purchase the core. The Xilinx CORE Generator system is bundled with the Xilinx ISE® Design Suite software at no additional charge.

Contact your local [Xilinx sales representative](#) for pricing and availability of Xilinx LogiCORE IP modules and software. Information about additional Xilinx LogiCORE IP modules is available on the Xilinx [IP Center](#).

Revision History

Date	Version	Revision
09/30/04	1.0	Initial Xilinx release.
04/28/05	2.0	Updated for Xilinx tools 7.1i SP1.
01/11/06	3.0	Updated for Xilinx tools 8.1i.
07/13/06	4.0	Updated to core version 8.0; Xilinx tools 8.2i.
09/21/06	4.1	Updated to core version 8.1, release date.
02/15/07	4.2	Updated to core version 8.2, Xilinx tools 9.1i.
08/08/07	4.3	Updated to core version 8.3, Xilinx tools 9.2i.
03/24/08	4.4	Updated references to IEEE Standard 802.3-2005, removed support for Virtex-II, updated tools for ISE v10.1.
06/27/08	8.5	Revision number advanced to match core version number, added footnote to facts table stating that WAN mode is no longer supported for Virtex-II Pro designs.
9/18/08	8.6	Updated to core version 8.6; CR fixes.
04/24/09	9.1	Updated to core version 9.1, Xilinx tools 11.1. Added Virtex-6 device support.
06/24/09	9.2	Updated to core version 9.2, Xilinx tools 11.2.
09/16/09	9.3	Updated to core version 9.3, Xilinx tools 11.3. Added Virtex-6 HXT, Virtex-6 -1L and Virtex-6 CXT support.
04/19/10	10.1	Updated to core version 10.1, Xilinx tools 12.1.
10/19/11	10.1.1	Documentation Only: Corrected data in Table 2.

Notice of Disclaimer

Xilinx is providing this product documentation, hereinafter “Information,” to you “AS IS” with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.