1-Gigabit Ethernet MAC v8.5

DS200 April 24, 2009 Product Specification

Introduction

The LogiCORE™ IP 1-Gigabit Ethernet Media Access Controller (GEMAC) core supports full-duplex operation at 1 Gigabit per second (Gbps), and can be used with all Gigabit Ethernet Physical Coding standards.

Features

- Designed to *IEEE 802.3-2005* specification
- Single-speed 1-Gbps Ethernet Media Access Controller (MAC)
- Full-duplex operation
  - Internal GMII physical-side interface (PHY) that can be connected to an embedded PHY core, such as the Xilinx Ethernet 1000BASE-X PCS/PMA or SGMII core or other custom logic
  - IOBs to provide an external GMII
  - A shim that includes DDRs and DCMs to provide an external RGMII
- Configured and monitored through an optional independent microprocessor-neutral interface
- Interfaces directly to the Xilinx Ethernet Statistics core for powerful statistics gathering
- Configurable flow control through MAC control pause frames; symmetrically or asymmetrically enabled
- Optional MDIO interface to managed objects in PHY layers (MII Management)
- Optional Address Filter with a selectable number of Address Table entries
- Support of VLAN frames to specification *IEEE 802.3-2005*
- Configurable support for jumbo frames of any length
- Configurable in-band FCS field passing on both transmit and receive paths
- Available under the terms of the SignOnce IP License

LogiCORE IP Facts	
Core Specifics	
Supported FPGA Family	**Speed Grade**
Virtex®-5	-1
Virtex-4	-10
Spartan®-3, Spartan-3E	-4
Spartan-3A/3AN/3A DSP[1]	-4
Performance	1 Gbps
Core Resources	
Slices	403-647[2] or 755-1307[3]
LUTs	626-936[2] or 769-1245[3]
FFs	650-1015[2] or 675-1074[3]
DCM	0-2[3]
BUFG	2-5[3]
Core Highlights	
Designed to *IEEE 802.3*	Simulation Only Evaluation
Hardware Verified	Hardware Evaluation
Provided with Core	
Documentation	Product Specification Getting Started Guide User Guide
Design File Formats	NGC Netlist, HDL Example Design, Demonstration Test Bench, Scripts
Constraints File	User Constraints File (.ucf)
Demo Example Designs	1-Gigabit Ethernet MAC with GMII 1-Gigabit Ethernet MAC with RGMII
Design Tool Requirements	
Supported HDL	VHDL and/or Verilog
Synthesis	XST 11.1
Xilinx Tools	ISE® v11.1
Simulation Tools	Mentor Graphics ModelSim v6.4b and above Cadence IUS v8.1-s009 and above Synopsys 2008.09[4] and above

1. See Table 19 for supported family configurations.
2. Virtex-5 FPGA slices and LUTs are different from previous families. See Tables 20 and 21.
3. See Tables 20 and 21; the precise number depends on user configuration.
4. Scripts provided for listed simulators only.

Applications

Typical applications for the GEMAC core include:

- Ethernet 1000BASE-X Port
- Ethernet 1000BASE-T Port

Ethernet 1000BASE-X Port

Figure 1 illustrates a typical GEMAC application. The PHY side of the core is connected to internally integrated RocketIO™ Multi-Gigabit Transceivers, available in certain families, to connect to an external off-the-shelf Gigabit Interface Converter (GBIC) or Small Form-Factor Pluggable (SFP) optical transceiver. The 1000BASE-X logic can be provided by the Ethernet 1000BASE-X PCS/PMA or SGMII cores.

The client side of the core is shown connected to the 10Mbps, 100 Mbps, 1 Gbps Ethernet FIFO, delivered with the GEMAC core to complete a single Gigabit Ethernet port. This port is shown connected to a Switch or Routing matrix, which may contain several ports.

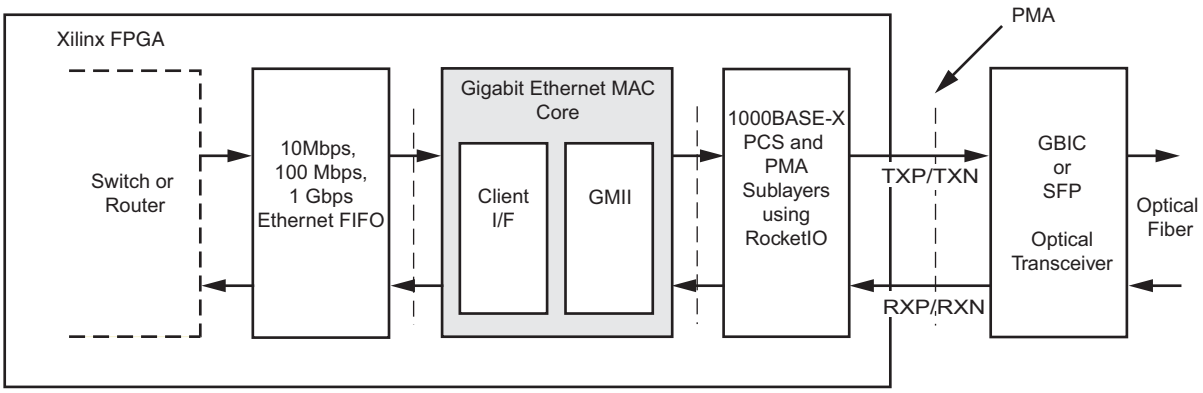

Figure 1: **Typical GEMAC 1000BASE-X Application**

Ethernet 1000BASE-T Port

Figure 2 illustrates a typical application for the GEMAC core. The PHY side of the core is implementing an external GMII by connecting it to IOBs. The external GMII is connected to an off-the-shelf Ethernet PHY device, which performs the 1000BASE-T standard. Alternatively, the external GMII may be replaced with an RGMII using a small logic shim. HDL example designs are provided with the core to demonstrate external GMII or RGMII.

The client side of the core is shown connected to the 10 Mbps, 100 Mbps, 1 Gbps Ethernet FIFO, delivered with the GEMAC core, to complete a single Gigabit Ethernet port. This port is shown connected to a Switch or Routing matrix, which may contain several ports.

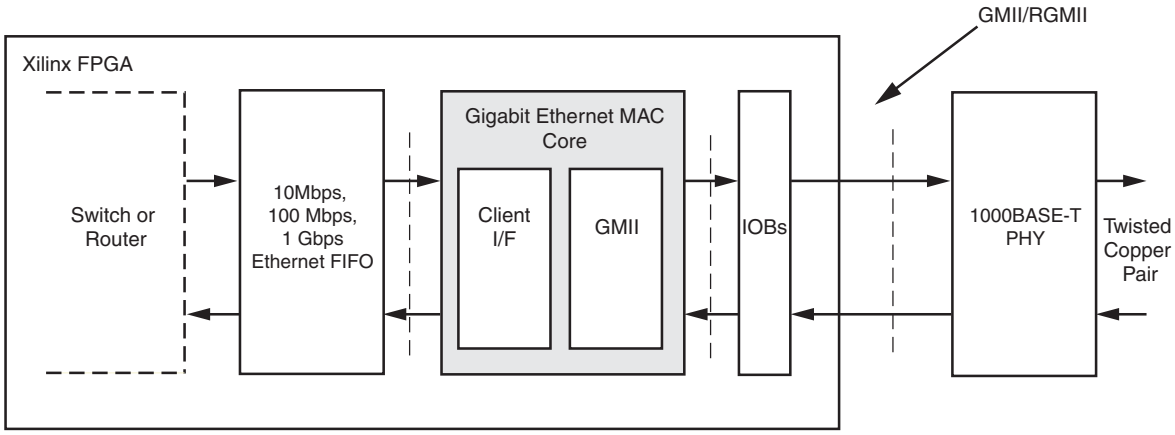

Figure 2: **Typical GEMAC 1000BASE-T Application**

Ethernet Architecture Overview

The GEMAC sublayer provided by this core is part of the Ethernet architecture illustrated in Figure 3. The part of this architecture from the MAC to the right is defined in *IEEE 802.3* specification. Figure 3 also illustrates where supported interfaces fit into the architecture.

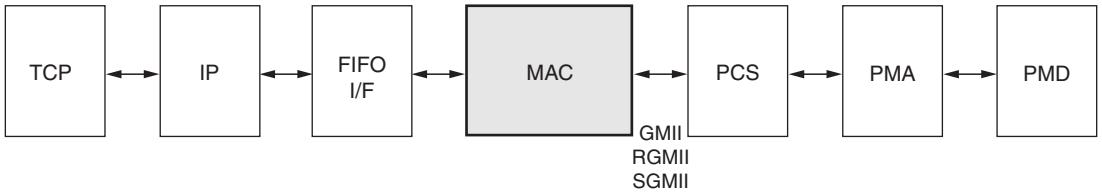

Figure 3: **Typical Ethernet Architecture**

MAC

The Ethernet Media Access Controller (MAC) is defined in the *IEEE 802.3* specification, in clauses 2, 3, and 4. A MAC is responsible for the Ethernet framing protocols and error detection of these frames. The MAC is independent of, and can connect to, any type of physical layer device (PHY).

GMII

The Gigabit Media Independent Interface (GMII) is defined in *IEEE 802.3*, clause 35. This is a parallel interface connecting a 1 Gigabit-capable MAC to the physical sublayers (PCS, PMA, and PMD).

RGMII

The Reduced Gigabit Media Independent Interface (RGMII) is an alternative to the GMII. RGMII achieves a 50% reduction in the pin count compared with GMII, and is therefore favored over GMII by PCB designers. This is achieved with the use of double-data-rate (DDR) flip-flops.

No change in the operation of the core is required to select GMII or RGMII. However, the clock management logic and IOB logic around the core does change. HDL example designs are provided with the core to implement either the GMII or RGMII protocols.

SGMII

The Serial-GMII (SGMII) is an alternative interface to the GMII that converts the parallel interface of the GMII into a serial format. This radically reduces the I/O count and is therefore often favored by PCB designers.

The GEMAC core can be extended to include SGMII functionality by internally connecting its PHY-side GMII to the Ethernet 1000BASE-X PCS/PMA or SGMII core. See the *1-Gigabit Ethernet MAC User Guide* for more information.

PCS, PMA, and PMD

The combination of the Physical Coding sublayer (PCS), the Physical Medium Attachment (PMA), and the Physical Medium Dependent (PMD) sublayer constitute the physical layers for the protocol. Two main physical standards are specified for Gigabit Ethernet:

- 1000BASE-X (defined in *IEEE 802.3*, clauses 36 to 39), provides short and long wavelength laser and short haul copper interfaces
- 1000BASE-T, (defined in *IEEE 802.3* clause 40), provides twisted-pair cabling systems

The 1000BASE-X architecture illustrated in Figure 1 can be provided by connecting the GEMAC core to the Ethernet 1000BASE-X PCS/PMA or SGMII core. See the *1-Gigabit Ethernet MAC User Guide* for details. The 1000BASE-T architecture illustrated in Figure 2 can be provided with the use of an external 1000BASE-T capable PHY device.

Core Overview

Figure 4 shows the major functional blocks and interfaces of the GEMAC core. Descriptions of these functional block and interfaces, along with associated signals are provided in the sections that follow.

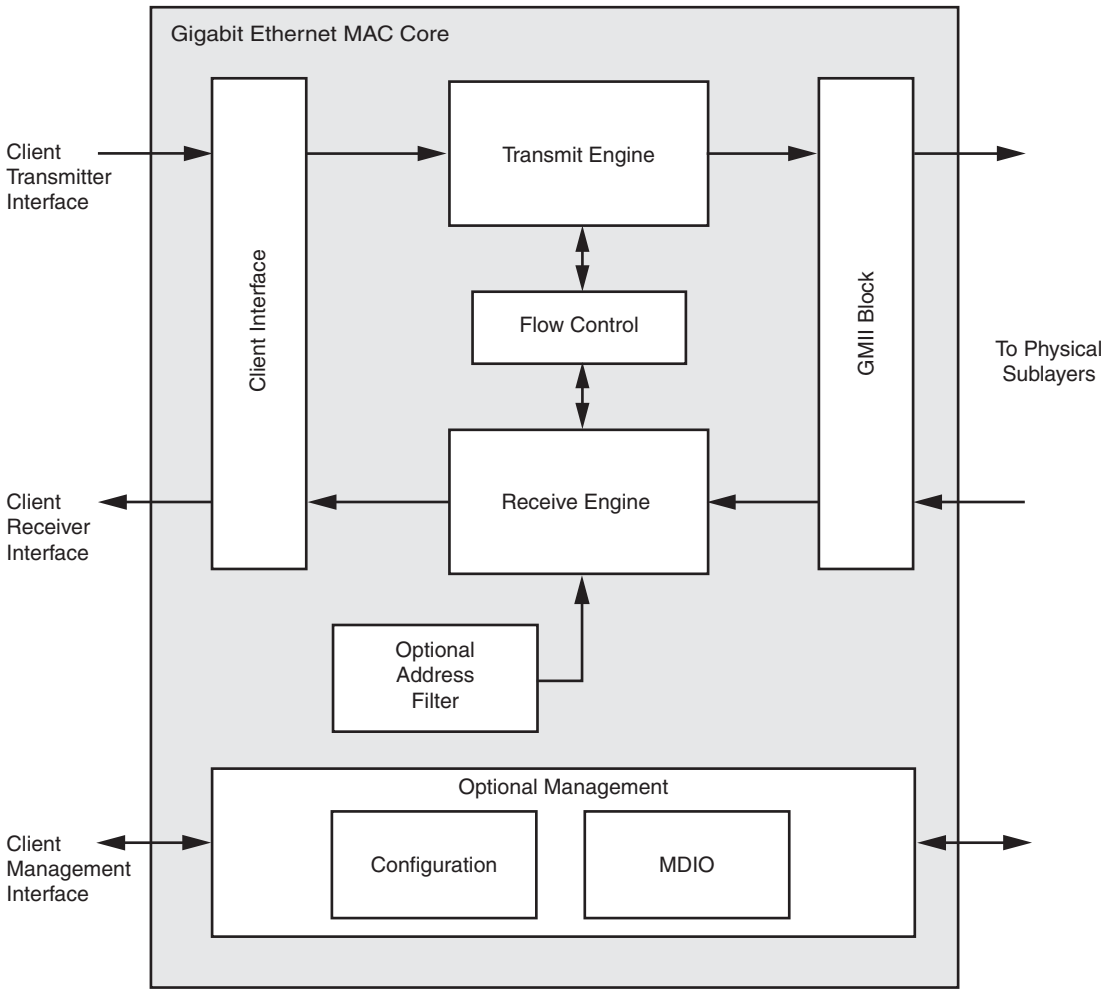

Figure 4: **GEMAC Functional Block Diagram**

Transmit Engine

The Transmit Engine accepts Ethernet frame data from the Client Transmitter Interface, adds the preamble field to the start of the frame, adds padding bytes if required (to ensure that the frame meets the minimum frame length requirements), and then adds the frame check sequence (when configured to do so). In addition, the transmitter is responsible for ensuring that the interframe spacing between successive frames always meets the minimum specified. The frame is then converted into a format compatible with the GMII and sent to the GMII Block.

Client Transmitter Interface Signals

Table 1 defines the GEMAC core client-side transmitter signals. These signals are used to transmit data from the client logic into the GEMAC core. See the *1-Gigabit Ethernet MAC User Guide* for more information.

Table 1: **Transmitter Client Interface Signal Pins**

Signal	Direction	Clock Domain	Description
gtx_clk	Input	n/a	Clock signal provided to the core at 125 MHz. Tolerance must be within *IEEE 802.3-2005* specification. This clock signal is used by all of the transmitter logic.
tx_data[7:0]	Input	gtx_clk	Frame data to be transmitted is supplied on this port.
tx_data_valid	Input	gtx_clk	Control signal for tx_data port.
tx_ifg_delay[7:0]	Input	gtx_clk	Control signal for configurable Inter Frame Gap adjustment.
tx_ack	Output	gtx_clk	Handshaking signal asserted when the current data on tx_data has been accepted.
tx_underrun	Input	gtx_clk	Asserted by client to force MAC core to corrupt the current frame.
tx_statistics_vector[31:0]	Output	gtx_clk	Provides statistical information on the last frame transmitted.
tx_statistics_valid	Output	gtx_clk	Asserted at end of frame transmission, indicating that the tx_statistics_vector is valid.

Client Transmitter Interface Operation

Figure 5 illustrates the timing of a normal outbound frame transfer. When the client initiates a frame transmission, it places the first column of data onto the `tx_data` port and asserts a logic 1 onto `tx_data_valid`.

After the GEMAC core reads the first byte of data, it asserts the `tx_ack` signal. On the next and subsequent rising clock edges, the client must provide the remainder of the data for the frame. The end of frame is signaled to the GEMAC core by taking `tx_data_valid` to logic 0.

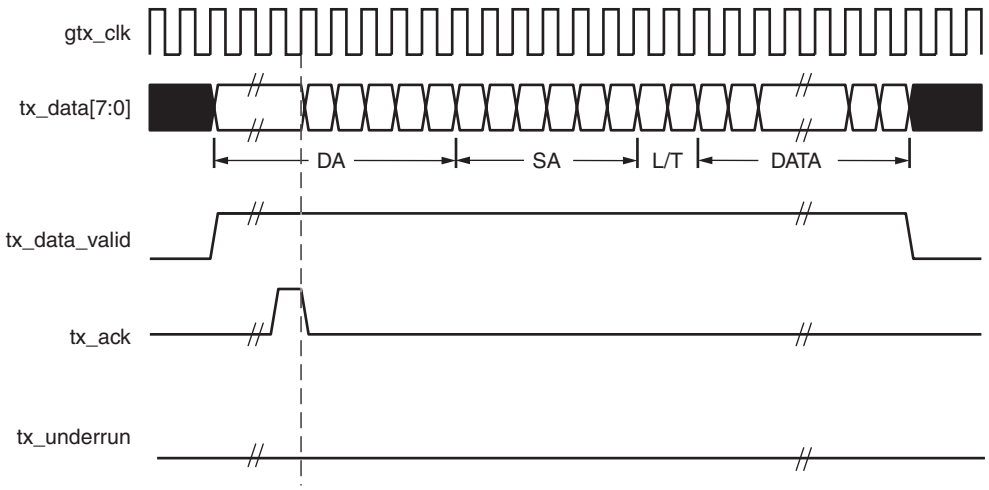

Figure 5: **Normal Frame Transmission Across Client Interface**

Receive Engine

The Receive Engine accepts Ethernet frame data from the GMII Block, removes the preamble field at the start of the frame, and removes padding bytes and frame check sequence (if required and when configured to do so). In addition, the receiver is responsible for performing error detection on the received frame using information that includes the frame check sequence field, received GMII error codes, and legal frame size boundaries.

Client Receiver Interface Signals

Table 2 defines the GEMAC core client-side receiver signals. These signals are used by the GEMAC core to transfer data to the client. For a complete description, see the *1-Gigabit Ethernet MAC User Guide*.

Table 2: **Receive Client Interface Signal Pins**

Signal	Direction	Clock Domain	Description
rx_data[7:0]	Output	gmii_rx_clk	Frame data received is supplied on this port.
rx_data_valid	Output	gmii_rx_clk	Control signal for the rx_data port.
rx_good_frame	Output	gmii_rx_clk	Asserted at end of frame reception to indicate that the frame should be processed by the MAC client.
rx_bad_frame	Output	gmii_rx_clk	Asserted at end of frame reception to indicate that the frame should be discarded by the MAC client.
rx_statistics_vector[27:0]	Output	gmii_rx_clk	This provides statistical information on the last frame received.
rx_statistics_valid	Output	gmii_rx_clk	Asserted at end of frame reception, indicating that the rx_statistics_vector is valid.

Client Receiver Interface Operation

Figure 6 illustrates the timing of a normal inbound frame transfer. The client must be prepared to accept data at any time; there is no buffering within the GEMAC to allow for latency in the receive client. After frame reception begins, data is transferred on consecutive clock cycles to the receive client until the frame is complete. The GEMAC asserts the `rx_good_frame` signal to indicate that the frame was successfully received and that the frame should be analyzed by the client.

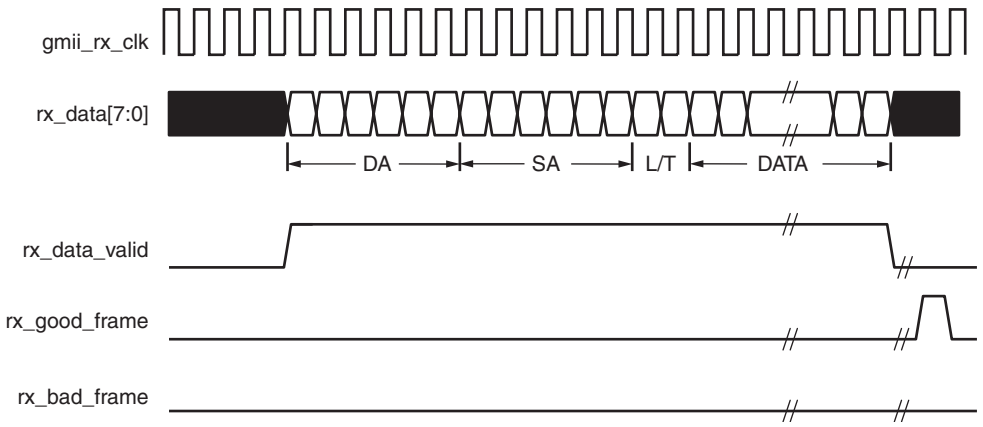

Figure 6: **Normal Frame Reception**

Flow Control

The Flow Control block is designed to clause 31 of the *IEEE 802.3-2005* standard. The GEMAC may be configured to send pause frames and to act upon their reception. These two behaviors can be configured asymmetrically. See the *1-Gigabit Ethernet MAC User Guide* for more information.

Flow Control Interface Signals

Table 3 defines the signals used by the client to request a flow-control action from the transmit engine.

Table 3: **Flow Control Interface Signal Pinout**

Signal	Direction	Clock Domain	Description
pause_req	Input	gtx_clk	**Pause request** Sends a pause frame down the link.
pause_val[15:0]	Input	gtx_clk	**Pause value** Inserted into the parameter field of the transmitted pause frame.

Transmitting a PAUSE Control Frame

The client initiates a Flow Control frame by asserting `pause_req` for a single clock period while the pause value is on the `pause_val[15:0]` bus. If the GEMAC core is configured to support transmit flow control, this action causes the core to transmit a PAUSE control frame on the link, with the PAUSE parameter set to the value on `pause_val[15:0]` in the cycle when `pause_req` was asserted. This does not disrupt any frame transmission in progress, but takes priority over any pending frame transmission. This frame is transmitted even if the transmitter is in a paused state.

Receiving a Pause Control Frame

When an error-free frame is received by the GEMAC core, it is evaluated in the following way:

1. The destination address field is matched against the MAC Control multicast address or the configured source address for the MAC.
2. The length/type field is matched against the MAC Control Type code.
3. If number 2 is true, the opcode field contents are matched against the PAUSE opcode.

If any of the previously listed conditions are false, or the MAC Flow Control logic for the receiver is disabled, the frame is ignored by the Flow Control logic and passed to the client with `rx_good_frame` asserted for interpretation.

If the frame passes all of the previously listed conditions, is of minimum legal size, and the MAC Flow Control logic for the receiver is enabled, the pause value parameter in the frame is used to inhibit transmitter operation after successful completion of the current packet transmission for the time defined in the *IEEE 802.3-2005* specification. Because the received pause frame has been acted on, it is passed to the client with `rx_bad_frame` asserted to indicate that it should be dropped.

Reception of any frame for which condition number 2 is true and is not of legal minimum length is considered an invalid control frame. This is ignored by the Flow Control logic and passed to the client with `rx_bad_frame` asserted.

Optional Address Filter

The GEMAC core can be implemented with an Address Filter. If the Address Filter is enabled, the device does not pass frames that do not contain one of a set of known addresses to the client.

The Address Filter can be programmed to respond to up to five user-defined addresses when the Management Interface is present in the core. These can be stored in a dedicated unicast address register and in a n-address deep table, where *n* is in the range 0 to 4. If the core is implemented with an Address Filter but the Management Interface is omitted from the core, only the unicast address register can be accessed. Access to the unicast address register is through the input signal `unicast_address[47:0]` when the Management Interface is not present.

In addition to the user-defined addresses, the broadcast and pause multicast addresses defined in the *IEEE 802.3-2005* and the pause frame MAC source address are also recognized. For a detailed description, see the 1-*Gigabit Ethernet MAC User Guide*.

Optional Management Interface

The Management Interface is an optional processor-independent interface with standard address, data, and control signals. It can be used as is, or a wrapper can be applied (not supplied) to interface to common bus architectures such as the CoreConnect bus interfacing to MicroBlaze™ or the embedded IBM PowerPC® processor (available in certain families). For a detailed description, see the 1-*Gigabit Ethernet MAC User Guide*.

Note:

This interface is used for the following:

- Configuration of the GEMAC core
- Access through the MDIO interface to the Management Registers located in the PHY connected to the GEMAC core

Client Management Interface Signals

Table 4 defines the optional signals used by the client to access the management features of the GEMAC core.

Table 4: **Optional Management Interface Signal Pinout**

Signal	Direction	Clock Domain	Description
host_clk	Input	n/a	Clock for the Management Interface; this must in the range of 10 MHz or above
host_opcode[1:0]	Input	host_clk	Defines operation to be performed over MDIO interface. Bit 1 is also used as a read/write control signal for configuration register access
host_addr[9:0]	Input	host_clk	Address of register to be accessed
host_wr_data[31:0]	Input	host_clk	Data to write to register
host_rd_data[31:0]	Output	host_clk	Data read from register
host_miim_sel	Input	host_clk	When asserted, the MDIO interface is accessed. When not asserted, the configuration registers are accessed
host_req	Input	host_clk	Used to signal a transaction on the MDIO interface
host_miim_rdy	Output	host_clk	When high, the MDIO interface has completed any pending transaction and is ready for a new transaction

Configuration Registers

After a power up or reset, the client can reconfigure the core parameters from the defaults. Configuration changes can be written at any time. Both the receiver and transmitter logic only respond to configuration changes during interframe gaps. The exceptions are the configurable resets, which take effect immediately.

Configuration of the GEMAC core is performed through a register bank that is accessed through the Management interface. Table 5 describes the available Configuration Registers. As described, the addresses have some implicit *don't care* bits; any access to an address in these performs a 32-bit read or write from the same configuration word.

Table 5: **Configuration Registers**

Address	Description
0x200-0x23F	Receiver Configuration (Word 0)
0x240-0x27F	Receiver Configuration (Word 1)
0x280-0x2BF	Transmitter Configuration
0x2C0-0x2FF	Flow Control Configuration
0x300-0x33F	Reserved
0x340-0x37F	Management Configuration
0x380-0x383	Unicast Address (Word 0) (if address filter is present)
0x384-0x387	Unicast Address (Word 1) (if address filter is present)

Table 5: **Configuration Registers *(Continued)***

Address	Description
0x388-0x38B	Address Table Configuration (Word 0) (if address filter is present)
0x38C-0x38F	Address Table Configuration (Word 1) (if address filter is present)
0x390-0x393	Address Filter Mode (if address filter is present)

Tables 6 and 7 define the register contents for the two receiver configuration words.

Table 6: **Receiver Configuration Word 0**

Bit	Default Value	Description
31-0	All 0s	Pause frame MAC Source Address[31:0]

Table 7: **Receiver Configuration Word 1**

Bit	Default Value	Description
15-0	All 0s	Pause frame MAC Source Address[47:32]
23-16	n/a	Reserved
24	0	Control Frame Length Check Disable
25	0	Length/Type Error Check Disable
26	n/a	Reserved
27	0	VLAN Enable
28	1	Receiver Enable
29	0	In-band FCS Enable
30	0	Jumbo Frame Enable
31	0	Receiver Reset

Table 8 defines the register contents for the Transmitter Configuration Word.

Table 8: **Transmitter Configuration Word**

Bit	Default Value	Description
24-0	n/a	Reserved
25	0	Interframe Gap Adjust Enable
26	n/a	Reserved
27	0	VLAN Enable
28	1	Transmit Enable
29	0	In-band FCS Enable
30	0	Jumbo Frame Enable
31	0	Transmitter Reset

Table 9 defines the register contents for the Flow Control Configuration Word.

Table 9: **Flow Control Configuration Word**

Bit	Default Value	Description
28-0	n/a	Reserved
29	1	Receiver Flow Control Enable
30	1	Transmitter Flow Control Enable
31	n/a	Reserved

Table 10 defines the register contents for the Management Configuration Word.

Table 10: **Management Configuration Word**

Bits	Default Value	Description
4-0	All 0s	Clock Divide[4:0]: This value enters a logical equation which enables the MDC frequency to be set as a divided down ratio of the HOST_CLK frequency.
5	0	MDIO Enable
31-6	n/a	Reserved

When the GEMAC core is implemented with an Address Filter, registers described in Tables 11 through 15 are used to access the Address Filter configuration. The register contents for the two unicast address registers are described in Tables 11 and 12.

Table 11: **Unicast Address (Word 0)**

Bits	Default Value	Description
31-0	All 0s	Address filter unicast address[31:0]

Table 12: **Unicast Address (Word 1)**

Bits	Default Value	Description
15-0	All 0s	Address filter unicast address[47:32]
31- 16	N/A	Reserved

Tables 13 and 14 show how the contents of the Address Table are set.

Table 13: **Address Table Configuration (Word 0)**

Bits	Default Value	Description
31-0	All 0s	MAC Address[31:0]

Table 14: **Address Table Configuration (Word 1)**

Bits	Default Value	Description
15-0	All 0s	MAC Address[47:32]
17-16	All 0s	The location in the address table that MAC address is to be read from or written to
22-18	N/A	Reserved
23	0	Read not write
31-24	N/A	Reserved

The contents of the Address Filter mode register are described in Table 15. If Promiscuous mode is set to 1, the Address Filter does not check the addresses of receive frames.

Table 15: **Address Filter Mode**

Bits	Default Value	Description
30-0	N/A	Reserved
31	0	Promiscuous Mode

MDIO Interface

The Management Interface is also used to access the MDIO interface of the GEMAC core; this interface is typically connected to the MDIO port of a PHY to access its configuration and status registers. The MDIO format is defined in *IEEE 802.3* clause 22.

MDIO Interface Signals

Table 16 defines the MDIO interface signals.

Table 16: **MDIO Interface Signal Pinout**

Signal	Direction	Clock Domain	Description
mdc	Output	host_clk	Management Clock: derived from host_clk on the basis of the Clock Divide[4:0] value in the Management Configuration Word.
mdio_in	Input	host_clk	Input data signal for communication with PHY configuration and status. Tie high if unused.
mdio_out	Output	host_clk	Output data signal for communication with PHY configuration and status.
mdio_tri	Output	host_clk	Tristate control for MDIO signals; 0 signals that the value on `mdio_out` should be asserted onto the MDIO bus.

Note: `mdio_in`, `mdio_out`, and `mdio_tri` can be connected to a Tri-state buffer to create a bi-directional `mdio` signal suitable for connection to an external PHY.

Configuration Vector

If the optional Management Interface is omitted from the GEMAC core, all relevant configuration settings described in Tables 6 through 9 and Table 15 are extracted as signals and bundled into the `configuration_vector[67:0]` signal. These signals can be permanently set by connecting to logic 0 or 1, or can be driven dynamically by control logic. See the *1-Gigabit Ethernet MAC User Guide*.

Reset Operation

The optional Management Interface provides independent configurable software driven resets for the receiver and transmitter paths (as defined in Tables 7 and 8.) When the Management interface is omitted, these resets are replaced as inputs of the `configuration_vector[64:0]` signal. In addition, a hardware reset port is provided to the core, described in Table 17.

Table 17: **Reset Interface Signal Pinout**

Signal	Direction	Clock Domain	Description
reset	Input	n/a	Asynchronous reset for the entire core. Active High.

GMII Block

This implements GMII-style signaling for the physical interface of the core and is typically attached to a PHY, either off-chip or internally integrated. The HDL example design delivered with the core when the GMII is selected connects these signals to IOBs to provide an external GMII. The HDL example design delivered with the core when the RGMII is selected connects these signals to a logic shim that uses double-data-rate (DDR) registers and DCMs to provide an external RGMII.

GMII Signals

Table 18 defines the GMII-side interface signals of the core.

Table 18: **GMII Interface Signal Pinout**

Signal	Direction	Clock Domain	Description
gmii_txd[7:0]	Output	gtx_clk	GMII Transmit data from MAC
gmii_tx_en	Output	gtx_clk	GMII Transmit control signal from MAC
gmii_tx_er	Output	gtx_clk	GMII Transmit control signal from MAC
gmii_rx_clk	Input	n/a	GMII Receive clock from an external PHY (125MHz)
gmii_rxd[7:0]	Input	gmii_rx_clk	GMII Received data to MAC
gmii_rx_dv	Input	gmii_rx_clk	GMII Received control signal to MAC
gmii_rx_er	Input	gmii_rx_clk	GMII Received control signal to MAC

Verification

The GEMAC core has been verified with extensive simulation and hardware testing, as detailed in this section.

Simulation

A highly parameterizable transaction-based test bench was used to test the core. Tests include:

- Register Access
- MDIO Access
- Frame Transmission and Error Handling
- Frame Reception and Error Handling
- Address Filtering

Hardware Verification

The GEMAC core has been tested in a variety of hardware test platforms at Xilinx to address specific parameterizations, including the following:

- The core has been tested with the Ethernet 1000BASE-X PCS/PMA or SGMII core, which follows the architecture illustrated in Figure 1. A test platform was built around these cores, including a back-end FIFO capable of performing a simple ping function and a test pattern generator. Software running on the embedded PowerPC processor was used to provide access to all configuration, status, and statistical counter registers. Version 3.0 of this core was taken to the University of New Hampshire Interoperability Lab (UNH IOL) where conformance and interoperability testing was performed.
- The core has been tested with an external 1000BASE-T PHY device, which follows the architecture illustrated in Figure 2. The GEMAC core was connected to the external PHY device using GMII, RGMII and SGMII (in conjunction with the Ethernet 1000BASE-X PCS/PMA or SGMII core).

Family Support

Table 19: **Family Support for the 1-Gigabit Ethernet MAC Core**

Device Family	PHY Interface		Management Interface	Address Filter
	with GMII	with RGMII		
Virtex-5	Supported	Supported	Supported	Supported
Virtex-4	Supported	Supported	Supported	Supported
Spartan-3	Supported	Supported	Supported	Supported
Spartan-3E	Supported	Not Supported	Supported	Supported
Spartan-3A/3AN/3A DSP	Supported	Supported	Supported	Supported

Device Utilization

The Virtex-5 device family contains six input LUTs; all other families contain four input LUTs. For this reason, the device utilization for Virtex-5 devices is listed separately. Please refer to either of the following:

- Virtex-5 Devices
- Other Device Families

Virtex-5 Devices

Table 20 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.

BUFG usage:

- does not consider multiple instantiations of the core, where clock resources can often be shared
- does not include the reference clock required for IDELAYCTRL. This clock source can be shared across the entire device and is not core specific

Table 20: **Device Utilization for Virtex-5 Device Families**

Parameter Values				Device Resources					
Physical Interface	Management Interface	Address Filter	Addr Table Entries	Slices	LUTs	FFs	18K Block RAMs	BUFGs	DCMs
GMII	Yes	Yes	4	647	933	1015	0	3	0[1]
GMII	Yes	Yes	0	568	835	959	0	3	0[1]
GMII	Yes	No	N/A	503	719	825	0	3	0[1]
GMII	No	Yes	N/A	404	675	702	0	2	0[1]
GMII	No	No	N/A	403	626	650	0	2	0[1]
RGMII	Yes	Yes	4	600	936	1015	0	3	0[1]
RGMII	Yes	Yes	0	568	835	959	0	3	0[1]
RGMII	Yes	No	N/A	517	722	825	0	3	0[1]
RGMII	No	Yes	N/A	394	678	702	0	2	0[1]
RGMII	No	No	N/A	361	629	650	0	2	0[1]

1. No core-specific DCMs are required if a reference clock for the IDELAYCTRL component is available on-chip.

Other Device Families

Table 21 provides approximate utilization figures for various core options when a single instance of the core is instantiated in a Spartan-3A device. Other families have similar utilization figures, except where stated.

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.

BUFG usage:

- does not consider multiple instantiations of the core, where clock resources can often be shared.
- does not include the reference clock required for any IDELAYCTRL component. This clock source can be shared across the entire device and is not core specific..

Table 21: **Device Utilization for Non-Virtex-5 Device Families**

Parameter Values				Device Resources					
Physical Interface	Management Interface	Address Filter	Addr Table Entries	Slices	LUTs	FFs	Block RAMs	BUFGs	DCMs
GMII	Yes	Yes	4	1307	1245	1039	0	3	1
GMII	Yes	Yes	0	1072	1048	983	0	3	1
GMII	Yes	No	N/A	975	920	849	0	3	1
GMII	No	Yes	N/A	841	821	727	0	2	1
GMII	No	No	N/A	755	769	675	0	2	1
RGMII	Yes	Yes	4	1257	1248	1074	0	4	2
RGMII	Yes	Yes	0	1155	1051	1018	0	4	2
RGMII	Yes	No	N/A	1042	923	884	0	4	2
RGMII	No	Yes	N/A	838	824	762	0	3	2
RGMII	No	No	N/A	812	772	710	0	3	2

References

[1] Virtex-4, and Virtex-5 FPGA User Guides

[2] *Spartan-3 Generation FPGA User Guide* (UG331)
(Information about the Spartan-3A DSP, Spartan-3AN, Spartan-3A, Spartan-3E, and Spartan-3 devices is provided.)

[3] *IEEE 802.3-2005* specification

Support

For technical support, visit www.xilinx.com/support/. Xilinx provides technical support for this product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not listed in the documentation, if customized beyond that allowed in the product documentation, or if any changes are made in sections of design marked *DO NOT MODIFY*.

Ordering Information

This Xilinx LogiCORE IP module is provided under the SignOnce IP Site License. Two free evaluation licenses are available. The Simulation Only license is provided with the CORE Generator™ and allows you to assess the core functionality and demonstrate the various interfaces on the core in simulation.

The Full System Hardware Evaluation license allows you to do the following:

- Fully integrate the core into an FPGA design
- Place and route the design
- Evaluate timing
- Perform back-annotated gate-level simulation of the core using the provided demonstration test bench
- Download and test the design in hardware for a limited period of time.

For access to all core functionality both in simulation and in hardware, you must purchase the GEMAC core. After purchasing, please go to the TEMAC product page (www.xilinx.com/products/ipcenter/GMAC.htm) for more information on generating your license key for use with the Xilinx Core Generator System v11.1.

Please contact your local Xilinx sales representative or visit the IP Center for information about additional Xilinx LogiCORE IP modules.

List of Acronyms

Acronym	Spelled Out
DCM	Digital Clock Manager
DDR	Double Data Rate
FCS	Frame Check Sequence
FIFO	First In First Out
FPGA	Field Programmable Gate Array.
GBIC	Gigabit Interface Converter
Gbps	Gigabit per second
GEMAC	Gigabit Ethernet Media Access Controller
GMII	Gigabit Media Independent Interface
HDL	Hardware Description Language
IP	Intellectual Property
MAC	Media Access Controller
MDIO	Management Data Input/Output
PCS	Physical Coding sublayer
PHY	physical-side interface
PMA	Physical Medium Attachment
PMD	Physical Medium Dependent
RGMII	Reduced Gigabit Media Independent Interface
SFP	Small Form-Factor Pluggable
SGMII	Serial Gigabit Media Independent Interface
VHDL	VHSIC Hardware Description Language (VHSIC an acronym for Very High-Speed Integrated Circuits).
VLAN	Virtual LAN (Local Area Network)

Revision History

Date	Version	Revision
9/24/04	5.0	Xilinx v5.0 release.
4/28/05	6.0	Xilinx v6.0 release; updated to Xilinx tools 7.1i and support for the Spartan-3E platform.
1/18/06	7.0	Xilinx v7.0 release; updated to Xilinx tools 8.1i and addition of address filter.
7/13/06	8.0	Xilinx v8.0 release; updated to Xilinx tools 8.2i.
9/21/06	8.1	Xilinx v8.1 release; support for Spartan-3A platform.
2/15/07	8.2	Xilinx v8.2 release; updated to Xilinx tools 9.1i.
08/08/07	9.0	Advanced core version to 8.3, updated supported tools for 9.2i release.
3/24/08	10.0	Updated core to version 8.4; Xilinx tools 10.1.
4/24/09	11.0	Updated core to version 8.5; Xilinx tools 11.1. Updated design tool requirements.
The product was discontinued as of August 31, 2009.		

Notice of Disclaimer