

## Introduction

The LogiCORE™ Tri-Mode Ethernet Media Access Controller (TEMAC) core supports half-duplex and full-duplex operation at 10 Mbps, 100 Mbps, and 1 Gbps.

## Features

- Designed to the *IEEE 802.3-2002* specification
- Configurable half-duplex and full-duplex operation
- Internal GMII physical-side (PHY) that can be connected to
  - An embedded PHY core, such as the LogiCORE Ethernet 1000BASE-X PCS/PMA or SGMII core or other custom logic
  - IOBs to provide an external GMII/MII
  - A shim to provide an external RGMII
- Configured and monitored through an optional independent microprocessor-neutral interface
- 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 selectable number of address table entries
- Optional clock enables to reduce clock resource usage
- Support of VLAN frames designed to *IEEE 802.3-2002*
- Configurable support of jumbo frames of any length
- Configurable interframe gap adjustment
- Configurable in-band FCS field passing on both transmit and receive paths
- Available under the terms of the [SignOnce IP Site License](#) agreement

LogiCORE Facts	
Core Specifics	
Supported Family	Virtex™-5, Virtex-4, Virtex-II, Virtex-II Pro, Spartan™-3, Spartan-3E, Spartan-3A/3AN/3A DSP
Speed Grade	<ul style="list-style-type: none"> <li>-1 for Virtex-5</li> <li>-10 for Virtex-4</li> <li>-5 for Virtex-II Pro</li> <li>-4 for Virtex-II, Spartan-3, Spartan-3E<sup>1</sup>, Spartan-3A/3AN/3A DSP</li> </ul>
Performance	10 Mbps, 100 Mbps, 1 Gbps
Core Resources	
Slices	577-930 <sup>2</sup> or 988-1568 <sup>4</sup>
LUTs	1072-1510 <sup>3</sup> or 1312-1977 <sup>4</sup>
FFs	1038-1548 <sup>3</sup> or 1024-1601 <sup>4</sup>
DCM	0-2 <sup>3</sup>
BUFG	2-6 <sup>4</sup>
Core Highlights	
Designed to IEEE specification <i>802.3-2002</i>	Hardware Verified
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)
Example Design	Tri-Mode Ethernet MAC with GMII/MII or RGMII interface
Demo Test Environment	
Design Tool Requirements	
Supported HDL	VHDL, Verilog®
Synthesis	XST 9.2i
Xilinx Tools	ISE™ 9.2i
Simulation tools	ModelSim® v6.1e Cadence™ IUS v5.8 <sup>4</sup>
Support	
Provided by Xilinx @ <a href="http://www.xilinx.com/support">www.xilinx.com/support</a>	

- Spartan-3E devices support only the GMII protocol.
- Virtex-5 slices and LUTs are different from previous families (see [Tables 24](#) and [25](#)).
- See [Tables 24](#) and [25](#); precise number depends on user configuration.
- Scripts provided for Mentor ModelSim and Cadence IUS only.

## Applications

Typical applications for the TEMAC core include the following:

- Ethernet Tri-Speed BASE-T Port (MII/GMII/RGMII)
- Ethernet Tri-Speed BASE-T Port (SGMII)

**Note:** The TEMAC core can be used in a 1000BASE-X port configuration by connecting the PHY side of the core to the Ethernet 1000BASE-X PCS/PMA or SGMII core; however, this creates a larger design than using the 1-Gigabit Ethernet MAC core.

### Ethernet Tri-Speed BASE-T Port (MII/GMII/RGMII)

**Figure 1** illustrates a typical application for the TEMAC core. The PHY side of the core is implementing an external GMII/MII by connecting it to IOBs. The external GMII/MII is connected to an off-the-shelf Ethernet PHY device, which performs the BASE-T standard at 1 Gbps, 100 Mbps, and 10 Mbps speeds. Alternatively, the external GMII/MII 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 TEMAC is shown connected to the 10 Mbps, 100 Mbps, 1 Gbps Ethernet FIFO (delivered with the example design) to complete a single Ethernet port. This port is shown connected to a Switch or Routing matrix, which may contain several ports.

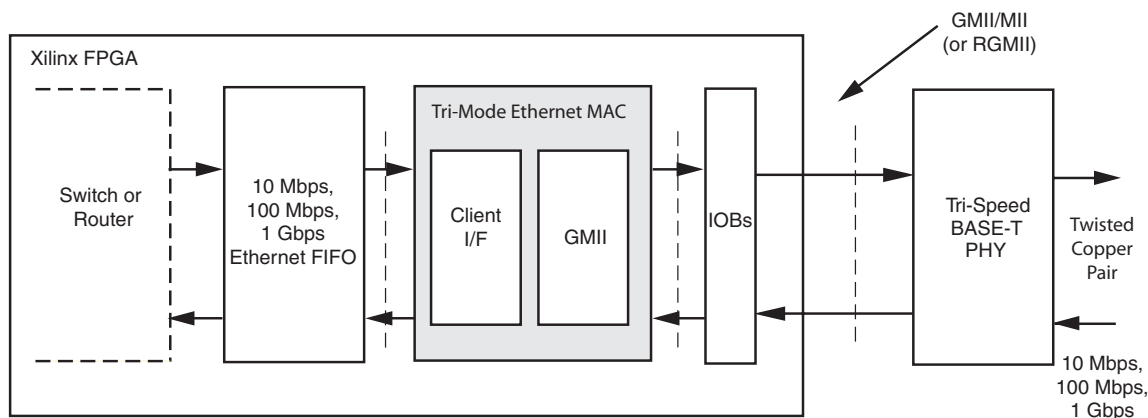


Figure 1: Typical BASE-T Application for TEMAC Core: MII/GMII/RGMII

## Ethernet Tri-Speed BASE-T Port (SGMII)

Figure 2 illustrates a typical application for the TEMAC core. The PHY side of the core is connected to internally integrated SGMII logic using the Virtex-II Pro RocketIO™ Multi-Gigabit Transceiver (MGT) to connect to an off-the-shelf Ethernet PHY device, which performs the BASE-T standard at 1 Gbps, 100 Mbps, and 10 Mbps speeds. The SGMII logic can be provided by the Ethernet 1000BASE-X PCS/PMA or SGMII core.

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

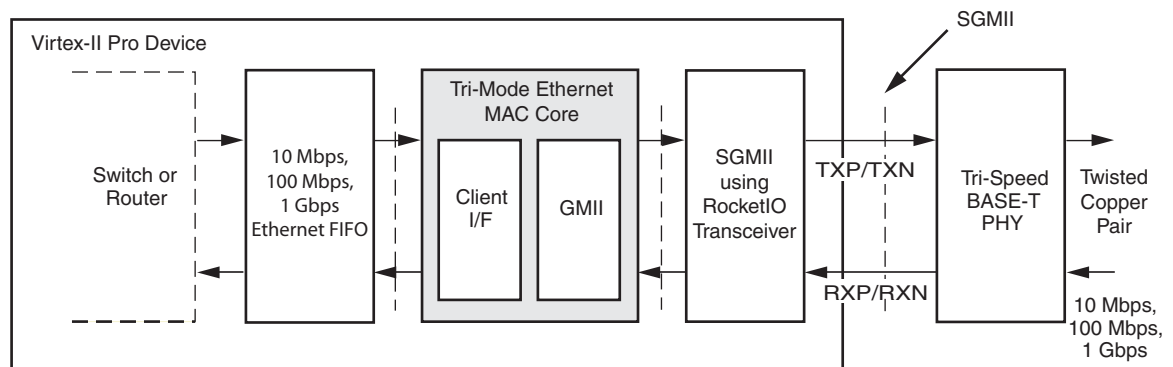


Figure 2: Typical BASE-T Application for TEMAC Core: SGMII

## Ethernet Architecture Overview

The TEMAC sublayer provided by this core is part of the Ethernet architecture displayed in [Figure 3](#). The portion of the architecture, from the MAC to the right, is defined in *IEEE 802.3*. This figure also illustrates where the supported interfaces fit into the architecture.

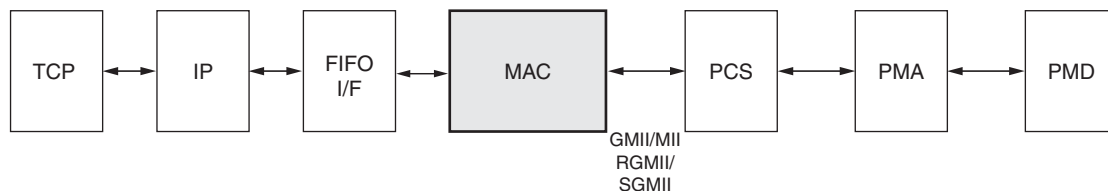


Figure 3: Typical Ethernet Architecture

### MAC

The Ethernet Medium Access Controller (MAC) is defined in *IEEE 802.3-2002* 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 be connected to, any type of physical layer.

### GMII / MII

The Gigabit Media Independent Interface (GMII) is defined in *IEEE 802.3-2002* clause 35. At 10 Mbps and 100 Mbps, the Media Independent Interface (MII) is used as defined in *IEEE 802.3-2002* clause 22. These are parallel interfaces connecting a 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 percent reduction in the pin count, compared with GMII, and for this reason is preferred 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 between GMII and RGMII. However, the clock management logic and IOB logic around the core will change. HDL example designs are provided with the core which implement either the GMII or RGMII protocols.

### SGMII

The Serial-GMII (SGMII) is an alternative interface to the GMII, which converts the parallel interface of the GMII into a serial format, radically reducing the I/O count (and for this reason often favored by PCB designers).

The TEMAC 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 from Xilinx. See the *Tri-Mode 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 comprise the physical layers of the Ethernet protocol.

Two main physical standards are specified for Ethernet:

- BASE-T, a copper standard using twisted pair cabling systems
- BASE-X, usually a fibre optical physical standard using short and long wavelength laser

BASE-T devices, supporting 10 Mbps, 100 Mbps, and 1 Gbps Ethernet speeds, are readily available as off-the-shelf parts. As illustrated in [Figure 1](#), these can be connected using GMII/MII, RGMII, or SGMII to provide a tri-speed Ethernet port.

The 1000BASE-X architecture illustrated in [Figure 2](#) can be provided by connecting the TEMAC core to the Ethernet 1000BASE-X PCS/PMA or SGMII core.

## Core Overview

[Figure 4](#) identifies the major functional blocks of the TEMAC core. Descriptions of the functional blocks and interfaces are provided in the subsequent sections.

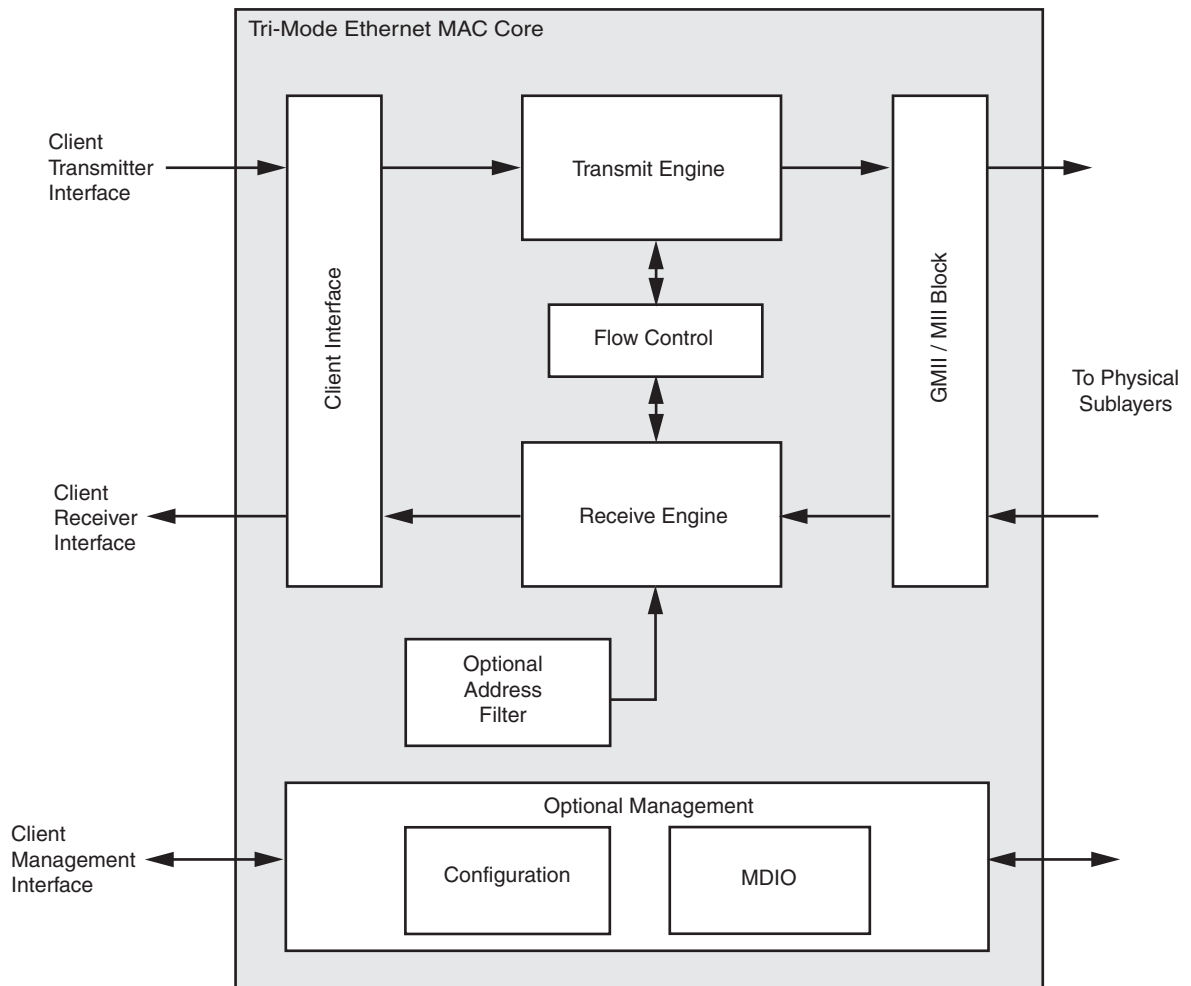


Figure 4: TEMAC Functional Block Diagram

## Client Interface

The client interface is designed for maximum flexibility in matching to a client switching fabric or network processor interface. The data pathway is 8-bits wide in both the transmit and receive directions. If the clock enable option is not selected, each pathway is synchronous to `txcoreclk` and `rxcoreclk` respectively. If the clock enable option is selected, each pathway is synchronous to `txgmiimiclk` and `rxgmiimiclk` respectively. In this mode of operation, transmit and receive enable inputs are driven to control the data throughput.

## Transmit Engine

The transmit engine takes data from the client and converts it to GMII format. Preamble and frame check sequence fields are added and the data is padded if necessary. The transmit engine also provides the transmit statistics vector for each packet and transmits the pause frames generated by the flow control module.

## Receive Engine

The receive engine takes the data from the GMII/MII interface and checks it for compliance to the *IEEE 802.3*. Padding fields are removed and the client is presented with the data along with a good or bad frame indicator. The receive engine also provides the receive statistics vector for each received packet.

## Flow Control

The flow control block is designed to *IEEE 802.3-2002* clause 31. The MAC can be configured to send pause frames with a programmable pause value and to act on their reception. These two behaviors can be configured asymmetrically.

## GMII/MII Block

The GMII/MII interface takes the data from the transmitter and converts it to MII format if the device is operating at speeds under 1 Gbps. The received data is converted into GMII format. At 1 Gbps, the data is simply passed through.

## Management Interface

The optional Management Interface is a processor-independent interface with standard address, data, and control signals. It is used for the configuration and monitoring of the TEMAC and for access to the MDIO Interface. It can be used as is or a wrapper can be applied (not supplied) to interface to common bus architectures. This interface is optional. If it is not present, the device can be configured using a configuration vector.

## MDIO Interface

The optional MDIO interface can be written to and read from using the Management Interface. The MDIO is used to monitor and configure PHY devices. The MDIO Interface is defined in *IEEE 802.3* clause 22.

## Address Filter

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

## Interface Descriptions

All ports of the core are internal connections in FPGA fabric. An example HDL design, provided in both VHDL and Verilog, is delivered with the core. The example design connects the core to a FIFO-based loopback example design and adds IBUFs, OBUFs, and IOB flip-flops to the external signals of the GMII/MII (or RGMII).

All clock management logic is placed in this example design, allowing the user more flexibility in implementation (for example, in designs using multiple cores). For information about the example design, see the *Tri-Mode Ethernet MAC Getting Started Guide*.

## Client Side Interface Signal Definition

### With Optional Clock Enables

**Table 1** defines the client-side transmit signals of the TEMAC core, which are used to transmit data from the client to the TEMAC core.

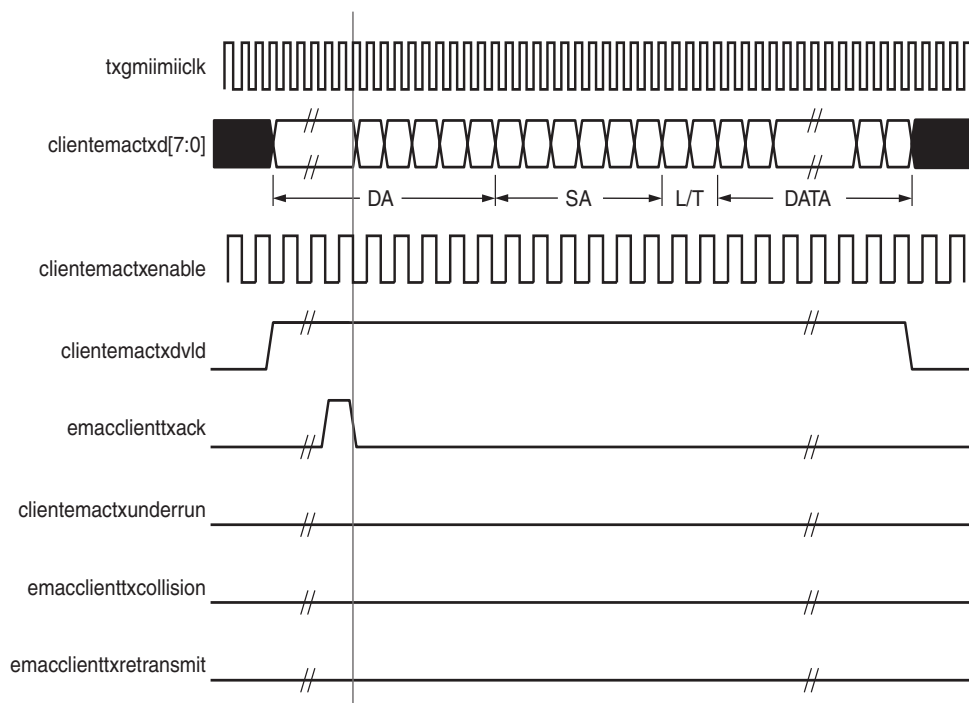
**Table 1: Transmit Client Interface Signal Pins (With Optional Clock Enables)**

Signal	Direction	Clock Domain	Description
clientemactxd[7:0]	Input	txgmiimiiclk	Frame data to be transmitted.
clientemactxdvld	Input	txgmiimiiclk	Control signal for clientemactxd port.
clientemactxenable	Input	txgmiimiiclk	Enable signal to the MAC core.
clientemactxifgdelay[7:0]	Input	txgmiimiiclk	Control signal for configurable interframe gap adjustment.
emacclienttxack	Output	txgmiimiiclk	Handshaking signal. Asserted when the current data on clientemactxd has been accepted.
clientemactxunderrun	Input	txgmiimiiclk	Asserted by client to force MAC core to corrupt the current frame.
emacclienttxcollision	Output	txgmiimiiclk	Asserted by the MAC core to signal a collision on the medium and that any transmission in progress should be aborted. Always 0 when the MAC core is in full-duplex mode.
emacclienttxretransmit	Output	txgmiimiiclk	When asserted at the same time as the emacclienttxcollision signal, this signals to the client that the aborted frame should be resupplied to the MAC core for retransmission. Always '0' when the MAC core is in full-duplex mode.
emacclienttxstats[31:0]	Output	txgmiimiiclk	A statistics vector that gives information on the last frame transmitted.
emacclienttxstatsvld	Output	txgmiimiiclk	Asserted at end of frame transmission, indicating that the emacclienttxstats is valid.

**Note:** All signals are active high.

**Figure 5** displays a typical frame transmission at the client interface at 10/100 Mbps, and **Figure 6** shows a typical frame transmission at 1 Gbps. The client asserts `clientemactxdvld` and puts the first byte of frame data on the `clientemactxd` bus. The client then waits until the TEMAC asserts `emacclienttxack` before sending the rest of the data. At the end of the frame, `clientemactxdvld` is deasserted.

At 1 Gbps, `clientemactxenable` should be set high. At 10/100 Mbps, `clientemactxenable` must be toggled on the rising edge of `txgmiimiiclk` to provide the correct data throughput in the MAC core. All transmit client logic should also be enabled using the `clientemactxenable` signal.



**Figure 5: Normal Frame Transmission at 10/100 Mbps with Clock Enables**



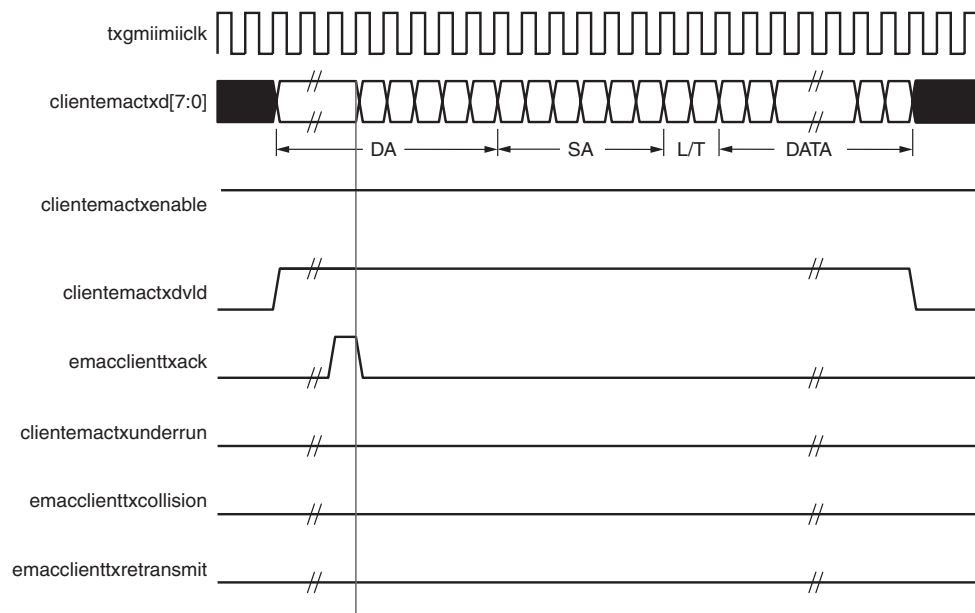


Figure 6: Normal Frame Transmission at 1 Gbps with Clock Enables

Table 2 describes the client-side receive signals used by the core to transfer data to the client.

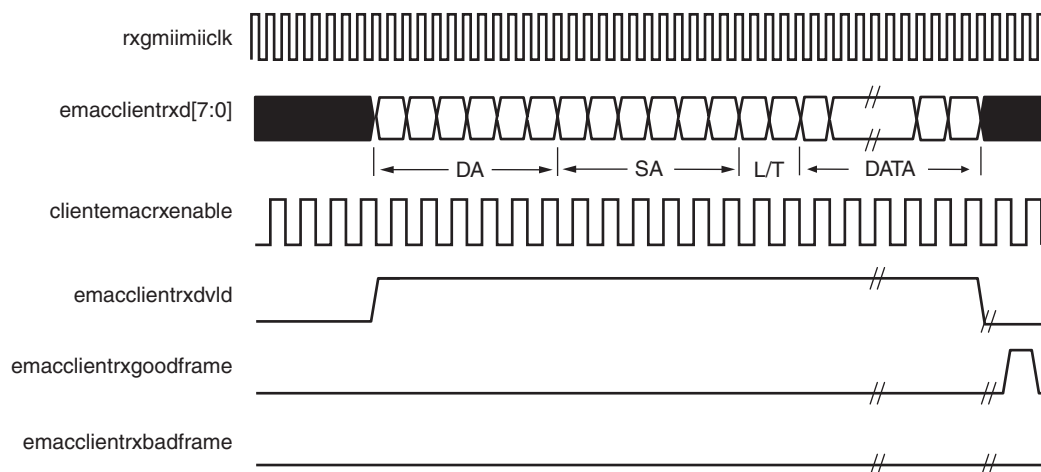
Table 2: Receive Client Interface Signal Pins (with Optional Clock Enables)

Signal	Direction	Clock Domain	Description
emacclientrx[7:0]	Output	rxgmiiiclk	Frame data received is supplied on this port.
emacclientrxvld	Output	rxgmiiiclk	Control signal for the emacclientrx port.
clientemacrxenable	Input	rxgmiiiclk	Enable signal for the received data.
emacclientrxgoodframe	Output	rxgmiiiclk	Asserted at end of frame reception to indicate that the frame should be processed by the MAC client.
emacclientrxbadframe	Output	rxgmiiiclk	Asserted at end of frame reception to indicate that the frame should be discarded by the MAC client.
emacclientrxstats[27:0]	Output	rxgmiiiclk	Provides information about the last frame received.
emacclientrxstatsvld	Output	rxgmiiiclk	Asserted at end of frame reception, indicating that the emacclientrxstats is valid.

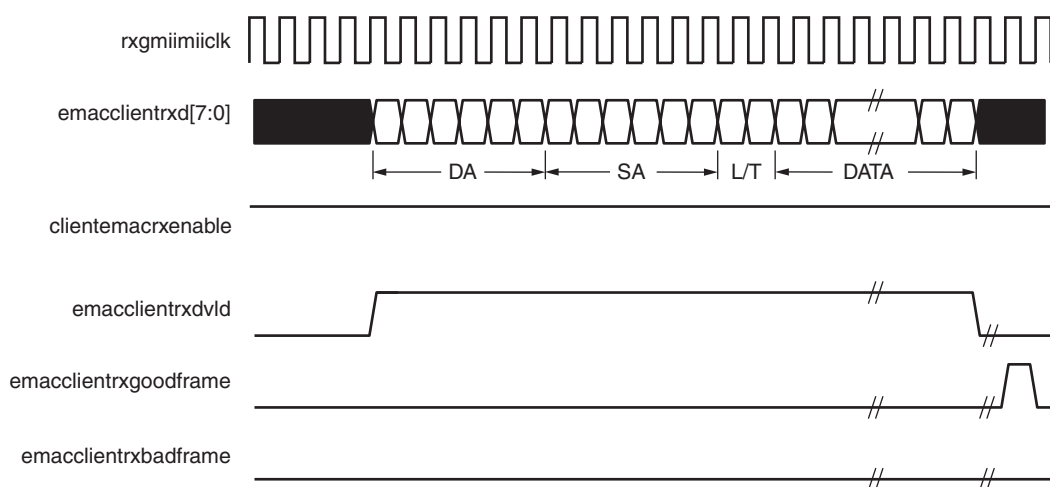
**Note:** All signals are active high.

Figure 7 displays the reception of a good frame at the client interface at 10/100 Mbps. Figure 8 shows the reception of a good frame at the client interface at 1 Gbps. The core asserts emacclientdvld for the duration of the frame data. At the end of the frame, the emacclientrxgoodframe signal is asserted to indicate that the frame passed all error checks

The receiver output is only valid when the `clientemacrxdenable` input is high. At 1 Gbps, `clientemacrxdenable` should be set high. At 10/100 Mbps, `clientemacrxdenable` must be toggled on the rising edge of `rxgmiimiiclk` to provide the correct data throughput in the TEMAC core. All receive client logic should also be enabled using the `clientemacrxdenable` signal.



**Figure 7: Normal Frame Reception at 10/100 Mbps with Clock Enables**



**Figure 8: Normal Frame Reception at 1 Gbps with Clock Enables**

## Without Optional Clock Enables

**Table 3** describes the client-side transmit signals of the MAC core when generated without the clock enable option. These signals are used to transmit data from the client to the MAC core.

**Table 3: Transmit Client Interface Signal Pins (Without Optional Clock Enables)**

Signal	Direction	Clock Domain	Description
clientemactxd[7:0]	Input	txcoreclk	Frame data to be transmitted.
clientemactxdvld	Input	txcoreclk	Control signal for clientemactxd port.
clientemactxifgdelay[7:0]	Input	txcoreclk	Control signal for configurable interframe gap adjustment.
emacclienttxack	Output	txcoreclk	Handshaking signal. Asserted when the current data on clientemactxd has been accepted.
clientemactxunderrun	Input	txcoreclk	Asserted by client to force MAC core to corrupt the current frame.
emacclienttxcollision	Output	txcoreclk	Asserted by the MAC core to signal a collision on the medium and that any transmission in progress should be aborted. Always 0 when the MAC core is in full-duplex mode.
emacclienttxretransmit	Output	txcoreclk	When asserted at the same time as the emacclienttxcollision signal, this signals to the client that the aborted frame should be resupplied to the MAC core for retransmission. Always '0' when the MAC core is in full-duplex mode.
emacclienttxstats[31:0]	Output	txcoreclk	A statistics vector that gives information on the last frame transmitted.
emacclienttxstatsvld	Output	txcoreclk	Asserted at end of frame transmission, indicating that the emacclienttxstats is valid.

**Note:** All signals are active high.

**Figure 9** displays a typical frame transmission at the client interface. As before, the client asserts `clientemactxdvld` and puts the first byte of frame data on the `clientemactxd` bus. The client then waits until the TEMAC asserts `emacclienttxack` before sending the rest of the data. At the end of the frame, `clientemactxdvld` is deasserted. These signals are now synchronous to the `txcoreclk` input.

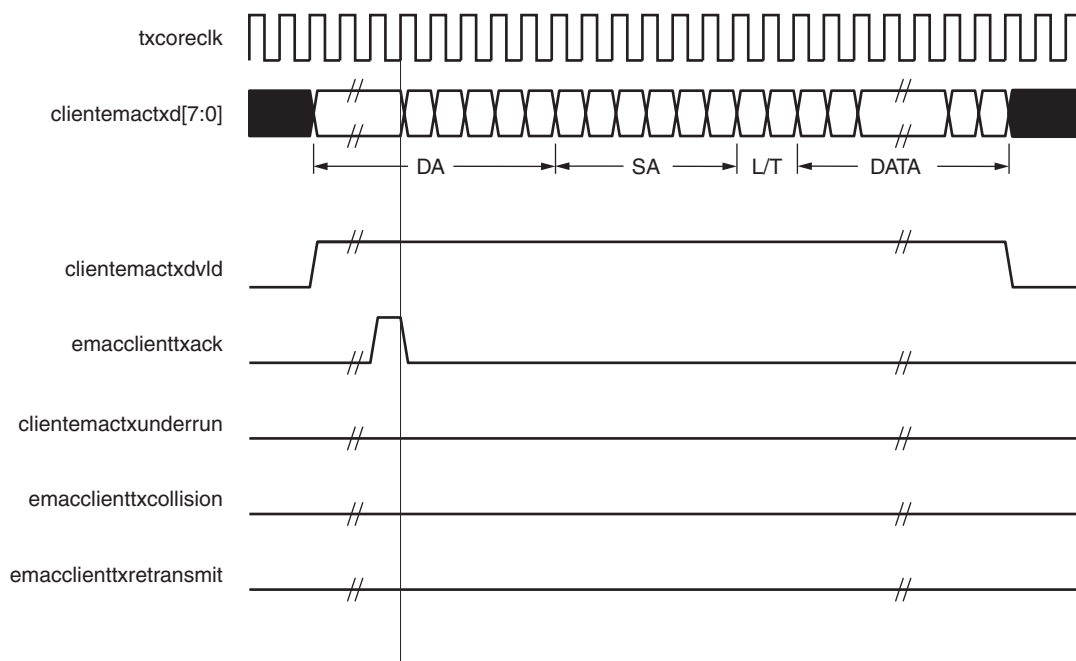


Figure 9: Normal Frame Transmission without Clock Enables

Table 4 describes the client-side receive signals used by the core to transfer data to the client.

Table 4: Receive Client Interface Signal Pins (Without Optional Clock Enables)

Signal	Direction	Clock Domain	Description
emacclientrx[7:0]	Output	rxcoreclk	Frame data received is supplied on this port.
emacclientrxvld	Output	rxcoreclk	Control signal for the emacclientrx port.
emacclientrxgoodframe	Output	rxcoreclk	Asserted at end of frame reception to indicate that the frame should be processed by the MAC client.
emacclientrxbadframe	Output	rxcoreclk	Asserted at end of frame reception to indicate that the frame should be discarded by the MAC client.
emacclientrxstats[27:0]	Output	rxcoreclk	Provides information about the last frame received.
emacclientrxstatsvld	Output	rxcoreclk	Asserted at end of frame reception, indicating that the emacclientrxstats is valid.

**Note:** All signals are active high.

Figure 10 displays the reception of a good frame at the client interface. The core asserts `emacclient-dvld` for the duration of the frame data. At the end of the frame, the `emacclientrxgoodframe` signal is asserted to indicate that the frame passed all error checks.

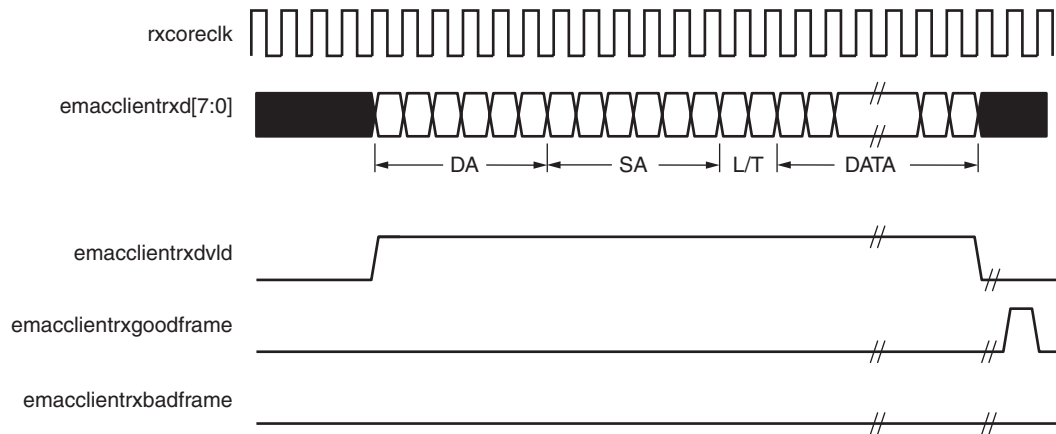


Figure 10: Normal Frame Reception without Clock Enables

Table 5 describes the signals used by the client to request a flow control action from the transmit engine. Valid flow control frames received by the MAC are automatically handled (if the MAC is configured to do so). The pause value in the received frame is used to inhibit the transmitter operation for the time defined in *IEEE 802.3-2002*. The frame is then passed to the client with `emacclientrxbadframe` asserted to indicate to the client that it should be dropped.

Table 5: Flow Control Interface Signal Pinout

Signal	Direction	Description
<code>clientemacpausereq</code>	Input	Pause request: Upon request the MAC transmits a pause frame upon the completion of the current data packet.
<code>clientemacpauseval[15:0]</code>	Input	Pause value: inserted into the parameter field of the transmitted pause frame.

**Note:** All signals are active high.

## Management Interface Signal Definition

Table 6 describes the optional signals used by the client to access the management features of the MAC core, including configuration, status and MDIO access.

Table 6: Optional Management Interface Signal Pinout

Signal	Direction	Clock Domain	Description
<code>hostclk</code>	Input	N/A	Clock for Management Interface.
<code>hostopcode[1:0]</code>	Input	<code>hostclk</code>	Defines operation to be performed over MDIO interface. Bit 1 is also used in configuration register access.
<code>hostaddr[9:0]</code>	Input	<code>hostclk</code>	Address of register to be accessed.
<code>hostwrdata[31:0]</code>	Input	<code>hostclk</code>	Data to write to register.
<code>hostrddata[31:0]</code>	Output	<code>hostclk</code>	Data read from register.

Table 6: Optional Management Interface Signal Pinout (*Continued*)

Signal	Direction	Clock Domain	Description
hostmiimsel	Input	hostclk	When asserted, the MDIO interface is accessed. When deasserted, the MAC internal configuration is accessed.
hostreq	Input	hostclk	Used to signal a transaction on the MDIO interface or to read from the statistic registers.
hostmiimrdy	Output	hostclk	When high, the MDIO interface has completed any pending transaction and is ready for a new transaction.

**Note:** All signals are active high.

## Configuration Registers

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

Configuration of the MAC core is performed through a register bank accessed through the Management Interface. The configuration registers available in the core are detailed in Table 7. As can be seen, the address has some implicit don't care bits; any access to an address in the ranges described will perform a 32-bit read or write from the same configuration word.

Table 7: 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
0x300-0x31F	MAC Speed Configuration
0x320-0x33F	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)
0x388-0x38B	Address Table Configuration (Word 0) (if address filter is present)
0x38C-0x38F	Address Table Configuration (Word 1) (if address filter is present)
0x390-0x3BF	Address Filter Mode (if address filter is present)

Tables 8 and 9 define the contents of the two receiver configuration words.

Table 8: Receiver Configuration Word 0

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

Table 9: Receiver Configuration Word 1

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

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

Table 10: Transmitter Configuration Word

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

Table 11 defines the register contents for the Flow Control Configuration Word.

Table 11: Flow Control Configuration Word

Bit	Default Value	Description
28-0	N/A	Reserved
29	1	Flow Control Enable (RX)
30	1	Flow Control Enable (TX)
31	N/A	Reserved

**Table 12** defines the register contents for the Management Configuration Word.

**Table 12: Management Configuration Word**

Bit	Default Value	Description
5-0	All 0s	Clock Divide[5:0].
6	0	MDIO Enable.
31-7	N/A	Reserved

**Table 13** defines the register contents for the MAC Speed Configuration Word.

**Table 13: MAC Speed Configuration Word**

Bit	Default Value	Description
29-0	N/A	Reserved
31-30	10	MAC Speed Configuration: <ul style="list-style-type: none"> <li>• 10 - 1 Gbps</li> <li>• 01 - 100 Mbps</li> <li>• 00 - 10 Mbps</li> </ul>

The address filter can be programmed to respond to up to 5 user-defined addresses. These can be stored in a dedicated unicast address register and, if the Management Interface is present, in a n-address deep table, where n can be in the range 0 to 4. In addition, the broadcast and pause multicast addresses defined in *IEEE 802.3-2002*, and the pause frame MAC source address (**Tables 8** and **9**) are also recognized. The register contents for the two unicast address registers are described in **Tables 14** and **15**.

**Table 14: Unicast Address (Word 0)**

Bit	Default Value	Description
31-0	tieemacunicastaddr[31 downto 0]	Address filter unicast address[31:0]

**Table 15: Unicast Address (Word 1)**

Bit	Default Value	Description
15-0	tieemacunicastaddr[47 downto 32]	Address filter unicast address[47:32]
31-16	N/A	Reserved

**Tables 16** and **17** show how the contents of the address table are set.

**Table 16: Address Table Configuration (Word 0)**

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



Table 17: Address Table Configuration (Word 1)

Bit	Default Value	Description
15-0	All 0s	MAC Address[47:32]
17-16	All 0s	The location in the address table (4 addresses deep) that the MAC address is to be written to or read from.
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 18. If promiscuous mode is set to '1,' the address filter does not check the addresses of the received frames.

Table 18: Address Filter Mode

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

## Configuration Vector Signal Definition

Table 19 describes the configuration vector, which uses direct inputs to the core to replace the functionality of the MAC configuration bits when the Management Interface is not used. The configuration settings described in Tables 8 through 11, Tables 13 through 15, and Table 18 are included in the vector. See the *Tri-Mode Ethernet MAC User Guide* for detailed information.

Table 19: Alternative to the Optional Management Interface: Configuration Vector Signal Pinout

Signal	Direction	Description
tieemacconfigvec[66:0]	Input	The Configuration Vector is used to replace the functionality of the MAC Configuration Registers when the Management Interface is not used.

**Note:** All bits of `tieemacconfigvec` are registered on input but can be treated as asynchronous inputs.

## Address Filter Signal Definition

Table 20 describes the address filter unicast address input.

Table 20: Address Filter Unicast Address

Signal	Direction	Description
tieemacunicastaddr[47:0]	Input	Vector used to set the default address for the MAC.

## Clock, Speed Indication, and Reset Signal Definition

**Table 21** describes the reset signal, the clock signals that are input to the core, and the outputs that can be used to select between the three operating speeds. The clock signals are generated in the top-level wrapper provided with the core.

**Table 21: Clock and Speed Indication Signals**

Signal	Direction	Description
reset	Input	Asynchronous reset for entire core.
txcoreclk	Input	Clock for data transmission on the client side of the core. 125 MHz at 1 Gbps, 12.5 MHz at 100 Mbps, and 1.25 MHz at 10 Mbps. This clock should be used to clock the client transmit circuitry. Only present if the core is not generated with the optional clock enable option.
rxcoreclk	Input	Clock for the reception of data on the client side of the core. 125 MHz at 1 Gbps, 12.5 MHz at 100 Mbps, and 1.25 MHz at 10 Mbps. This clock should be used to clock the client receiver circuitry. Only present if the core is not generated with the optional clock enable option.
txgmiimiiclk	Input	Clock for the transmission of data on the physical interface. 125 MHz at 1 Gbps, 25 MHz at 100 Mbps, and 2.5 MHz at 10 Mbps. This clock should be used to clock the physical interface transmit circuitry. If the core is generated with the clock enable option, this clock is also used to clock the client transmit circuitry.
rxgmiimiiclk	Input	Clock for the reception of data on the physical interface. 125 MHz at 1 Gbps, 25 MHz at 100 Mbps, and 2.5 MHz at 10 Mbps. This clock should be used to clock the physical interface receive circuitry. If the core is generated with the clock enable option, this clock is also used to clock the client receiver circuitry.
speedis100	Output	Output asserted when the core is operating at 100 Mbps. It is derived from a configuration register (if the optional Management Interface is present) or from the tieemacconfigvec configuration vector (if the optional Management Interface is not present).
speedis10100	Output	This output is asserted when the core is operating at either 10 Mbps or 100 Mbps. It is derived from a configuration register (if the optional Management Interface is present) or from the tieemacconfigvec configuration vector (if the optional Management Interface is not present).

## Physical Interface Signal Definition

**Table 22** describes the MDIO (MII Management) interface signals of the TEMAC core, which are typically connected to the MDIO port of a PHY device, either off-chip or an SoC-integrated core. The MDIO format is defined in *IEEE 802.3-2002* clause 22.

**Table 22: MDIO Interface Signal Pinout**

Signal	Direction	Description
emacphymclkout	Output	MDIO Management Clock: derived from hostclk on the basis of supplied configuration data when the optional Management Interface is used.
emacphymdin	Input	Input data signal for communication with PHY configuration and status. Tie high if unused.
emacphymdout	Output	Output data signal for communication with PHY configuration and status.
emacphymdtri	Output	Tristate control for MDIO signals; '0' signals that the value on MDIO_OUT should be asserted onto the MDIO bus.

**Table 23** describes the GMII/MII signals of the TEMAC core, which are typically attached to a PHY module, either off-chip or internally integrated. The GMII is defined in *IEEE 802.3-2002* clause 35, and MII is defined in *IEEE 802.3-2002* clause 22.

**Table 23: Optional GMII Interface Signal Pinout**

Signal	Direction	Clock Domain	Description
emacphytxd[7:0]	Output	gmiimiitxclk	Transmit data to PHY
emacphytxen	Output	gmiimiitxclk	Data Enable control signal to PHY
emacphytxer	Output	gmiimiitxclk	Error control signal to PHY
phyemaccrs	Input	N/A	Control signal from PHY
phyemaccol	Input	N/A	Control signal from PHY
phyemacrxd[7:0]	Input	gmiimiirxclk	Received data from PHY
phyemacrxdv	Input	gmiimiirxclk	Data Valid control signal from PHY
phyemacrxfcr	Input	gmiimiirxclk	Error control signal from PHY

## Verification

The TEMAC 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 TEMAC 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 Xilinx Ethernet 1000BASE-X PCS/PMA or SGMII core, as illustrated in the architecture displayed in [Figure 2](#). 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<sup>™</sup> was used to provide access to all configuration, status, and statistical counter registers. Conformance and interoperability testing was performed at the University of New Hampshire Interoperability Lab (UNH IOL) on version 1.1 of the TEMAC core.
- The core has been tested with an external 1000BASE-T PHY device, as illustrated in the architecture displayed in [Figure 1](#). The TEMAC 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).

## Device Utilization

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

- **Virtex-5**
- **Other Device Families (not Virtex-5)**

### Virtex-5

**Table 24** 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 24: Device Utilization for Virtex-5 Families**

Core Parameters					Device Resources					
Physical Interface	Management Interface	Address Filter	Clock Enable	Addr Table Entries	Slices	LUTs	FFs	18k Block RAMs	BUFGs	DCM
GMII	Yes	Yes	No	4	930	1497	1541	2	5	0 <sup>1</sup>
GMII	Yes	No	No	N/A	768	1229	1349	2	5	0 <sup>1</sup>
GMII	No	Yes	No	0	659	1177	1227	2	4	0 <sup>1</sup>
GMII	No	No	No	N/A	695	1127	1173	2	4	0 <sup>1</sup>
GMII	Yes	Yes	Yes	4	819	1448	1406	0	3	0 <sup>1</sup>
GMII	Yes	No	Yes	N/A	691	1164	1214	0	3	0 <sup>1</sup>
GMII	No	Yes	Yes	0	603	1131	1092	0	2	0 <sup>1</sup>
GMII	No	No	Yes	N/A	577	1072	1038	0	2	0 <sup>1</sup>
RGMII	Yes	Yes	No	4	924	1510	1548	2	5	0 <sup>1</sup>
RGMII	Yes	No	No	N/A	800	1242	1356	2	5	0 <sup>1</sup>
RGMII	No	Yes	No	0	716	1191	1234	2	4	0 <sup>1</sup>
RGMII	No	No	No	N/A	677	1140	1180	2	4	0 <sup>1</sup>
RGMII	Yes	Yes	Yes	4	821	1461	1413	0	3	0 <sup>1</sup>
RGMII	Yes	No	Yes	N/A	723	1178	1221	0	3	0 <sup>1</sup>
RGMII	No	Yes	Yes	0	650	1144	1099	0	2	0 <sup>1</sup>
RGMII	No	No	Yes	N/A	637	1086	1045	0	2	0 <sup>1</sup>

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

## Other Device Families (not Virtex-5)

**Table 25** provides approximate utilization figures for various core options when a single instance of the core is instantiated in a Virtex-II Pro 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 25: Device Utilization for non-Virtex-5 Families**

Core Parameters					Device Resources					
Physical Interface	Management Interface	Address Filter	Clock Enable	Addr Table Entries	Slices	LUTs	FFs	Block RAMs	BUFGs	DCM
GMII	Yes	Yes	No	4	1514	1957	1527	2	5	0 <sup>1,2</sup>
GMII	Yes	No	No	N/A	1272	1584	1335	2	5	0 <sup>1,2</sup>
GMII	No	Yes	No	0	1140	1478	1212	2	4	0 <sup>1,2</sup>
GMII	No	No	No	N/A	1107	1425	1158	2	4	0 <sup>1,2</sup>
GMII	Yes	Yes	Yes	4	1394	1862	1392	0	3	0 <sup>1,2</sup>
GMII	Yes	No	Yes	N/A	1151	1474	1200	0	3	0 <sup>1,2</sup>
GMII	No	Yes	Yes	0	1022	1374	1078	0	2	0 <sup>1,2</sup>
GMII	No	No	Yes	N/A	988	1312	1024	0	2	0 <sup>1,2</sup>
RGMII	Yes	Yes	No	4	1568	1977	1601	2	6	2 <sup>2,3</sup>
RGMII	Yes	No	No	N/A	1326	1604	1409	2	6	2 <sup>2,3</sup>
RGMII	No	Yes	No	0	1195	1498	1286	2	5	2 <sup>2,3</sup>
RGMII	No	No	No	N/A	1162	1445	1232	2	5	2 <sup>2,3</sup>
RGMII	Yes	Yes	Yes	4	1450	1881	1466	0	4	2 <sup>2,3</sup>
RGMII	Yes	No	Yes	N/A	1206	1493	1274	0	4	2 <sup>2,3</sup>
RGMII	No	Yes	Yes	0	1076	1393	1152	0	3	2 <sup>2,3</sup>
RGMII	No	No	Yes	N/A	1042	1331	1098	0	3	2 <sup>2,3</sup>

1. Spartan-3, Spartan-3E and Spartan-3A device families require one DCM with external GMII to meet input setup and hold timing on the receiver path.
2. For Virtex-4 device families, this does not include any DCMs that may be required to generate the reference clock for the IDELAYCTRL component, as this clock source can be shared across the entire device and is not core specific.
3. Virtex-4 device families only require one DCM with external RGMII, as IDELAY components can be used to meet input setup and hold timing on the receiver path.

## References

- [1] Virtex-II, Virtex-II Pro, and Virtex-4, and Virtex-5 User Guides
- [2] Spartan-3, Spartan-3E, Spartan-3A, Spartan-3AN, Spartan-3A DSP Data Sheets
- [3] *IEEE 802.3-2002* specification

## Support

For technical support, visit [www.xilinx.com/support](http://www.xilinx.com/support). Xilinx provides technical support for this LogiCORE 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 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™, 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 [TEMAC product page](#).

For full access to all core functionality, both in simulation and in hardware, you must purchase the core. After purchase, the core can be downloaded from the Xilinx [IP Center](#) for use with the Xilinx CORE Generator System v9.2i. The Xilinx CORE Generator system is bundled with the Xilinx ISE Foundation software at no additional charge.

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

## Revision History

The following table provides the document revision history.

Date	Version	Revision
7/06/2004	1.0	Initial draft.
9/24/04	1.1	Initial Xilinx release.
4/28/05	2.0	Updated to version 2.1 of the core, Xilinx tools v7.1i, and support for Spartan-3E.
1/18/06	2.1	Updated to release date, version 2.2 of the core, and Xilinx tools 8.1i
7/13/06	3.1	Core version 3.1; Xilinx tools 8.2i.
9/21/06	3.2	Core version updated to 3.2, Spartan-3A added to supported device family in Facts table.
2/15/07	3.3	Updated to version 3.3; Xilinx tools 9.1i.
8/8/07	3.4	Updated core to version 3.4; Xilinx tools v9.2i, Cadence IUS v5.8.