Virtex-4 Tri-Mode Embedded Ethernet MAC Wrapper v4.4

DS307 February 15, 2007 Product Specification

Introduction

The LogiCORE™ Virtex-4™ Embedded Tri-Mode Ethernet Media Access Controller (MAC) Wrapper automates the generation of HDL wrapper files for the Tri-Mode Ethernet MAC in Virtex-4 FX devices using the Xilinx CORE Generator™.

VHDL and Verilog instantiation templates are available in the Libraries Guide for the Virtex-4 Ethernet MAC primitive; however, due to the complexity and the large number of ports, the CORE Generator simplifies integration of the Ethernet MAC by providing HDL examples based on user-selectable configurations.

Features

- Allows selection of one or both Ethernet MACs (EMAC0/EMAC1) from the Embedded Ethernet MAC primitive
- Connects the EMAC0/EMAC1 tie-off pins based on user options
- Provides user-configurable Ethernet MAC physical interfaces, including
  - Supports MII, GMII, RGMII v1.3, RGMII v2.0, SGMII, and 1000BASE-X PCS/PMA interfaces
  - Instantiates clock buffers, DCMs, RocketIO™ Multi-Gigabit Transceivers (MGTs), and logic as required for the selected physical interfaces
- Provides a simple FIFO-loopback example design, which is connected to the MAC client interfaces
- Provides a simple demonstration test bench based on the selected configuration
- Includes an example of a low-level driver for DCR accesses
- Generates VHDL or Verilog

LogiCORE Facts	
Supported Family	Virtex-4 FX
Performance	10 Mbps, 100 Mbps, 1 Gbps
Example Design Resources	
Slices	366-1112[1]
LUTs	420-1233[1]
FFs	432-1355[1]
Block RAMs	4-8[1]
DCM	0-2[1]
BUFG	2-8[1]
Wrapper Highlights	
Optimized Clocking Logic	HDL Example Design
Hardware Verified	Demonstration Test Bench
Provided with Wrapper	
Documentation	Product Specification Getting Started Guide User Guide[2]
Design File Formats	HDL Example Design, Demonstration Test Bench, Scripts
Constraints File	User Constraints File (UCF)
Example Designs	Example FIFO connected to client I/F
	Demonstration Test Environment
Design Tool Requirements	
Supported HDL	VHDL and/or Verilog
Synthesis	XST 9.1i
Xilinx Tools	ISE™ 9.1i
Simulation Tools (SWIFT-compliant simulator required)	Mentor ModelSim® 6.1e Cadence™ IUS[3]

1. The precise number depends on user configuration; see "Device Utilization" on page 7.
2. The *Virtex-4 Embedded Tri-Mode Ethernet MAC User Guide* is available under the Related Information area of the product page.
3. Scripts provided for Mentor ModelSim and Cadence IUS only.

Ethernet Architecture Overview

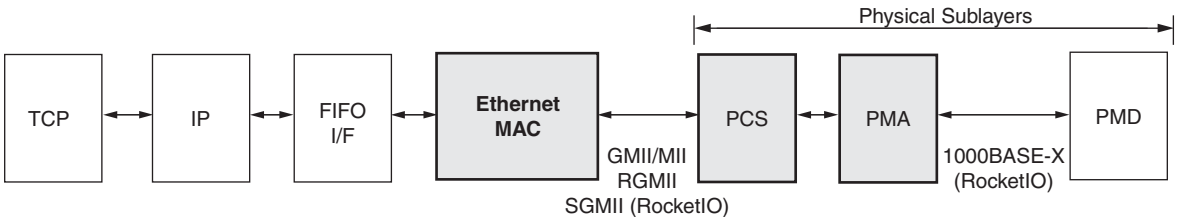

Figure 1: **Typical Ethernet Architecture**

Figure 1 displays the Ethernet MAC architecture from the MAC to the right, as defined in the *IEEE 802.3* specification, and also illustrates where the supported physical interfaces fit into the architecture.

MAC

The Ethernet MAC is defined in the *IEEE 802.3* specification 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 sublayer.

GMII/MII

The Media Independent Interface (MII), defined in *IEEE 802.3* clause 22, is a parallel interface that connects a 10-Mbps and/or 100-Mbps capable MAC to the physical sublayers. The Gigabit Media Independent Interface (GMII), defined in *IEEE 802.3* clause 35, is an extension of the MII used to connect a 1-Gbps capable MAC to the physical sublayers. MII can be considered a subset of GMII, and as a result, GMII/MII can carry Ethernet traffic at 10 Mbps, 100 Mbps, and 1 Gbps.

RGMII

The Reduced-GMII (RGMII) is an alternative to the GMII/MII. RGMII achieves a 45 percent reduction in the pin count, achieved by the use of double-data-rate (DDR) flip-flops. For this reason, RGMII is preferred over GMII by PCB designers. RGMII can carry Ethernet traffic at 10 Mbps, 100 Mbps, and 1 Gbps.

SGMII

The Serial-GMII (SGMII) interface is an alternative to the GMII/MII. SGMII converts the parallel interface of the GMII/MII into a serial format using a RocketIO, radically reducing the I/O count. For this reason, it is often the preferred interface of PCB designers. SGMII can carry Ethernet traffic at 10 Mbps, 100 Mbps, and 1 Gbps.

PCS, PMA, 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 Figures 1 and 2, these can be connected using GMII/MII, RGMII, or SGMII to provide a tri-speed Ethernet port.

The Ethernet MAC has built-in 1000BASE-X PCS/PMA functionality and can be connected to a RocketIO to provide a 1 Gbps fibre optic port, as illustrated in Figure 3.

Applications

Typical applications for the Ethernet MAC core include

- Ethernet Tri-Speed BASE-T Port
- Ethernet 1000BASE-X Port

Ethernet Tri-Speed BASE-T Port

Figure 2 illustrates a typical application for a single Ethernet MAC. 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 can be replaced with an RGMII (as shown) or as an SGMII (which requires the use of a RocketIO). GMII, RGMII, and SGMII functionality are demonstrated in the HDL examples provided with the example design.

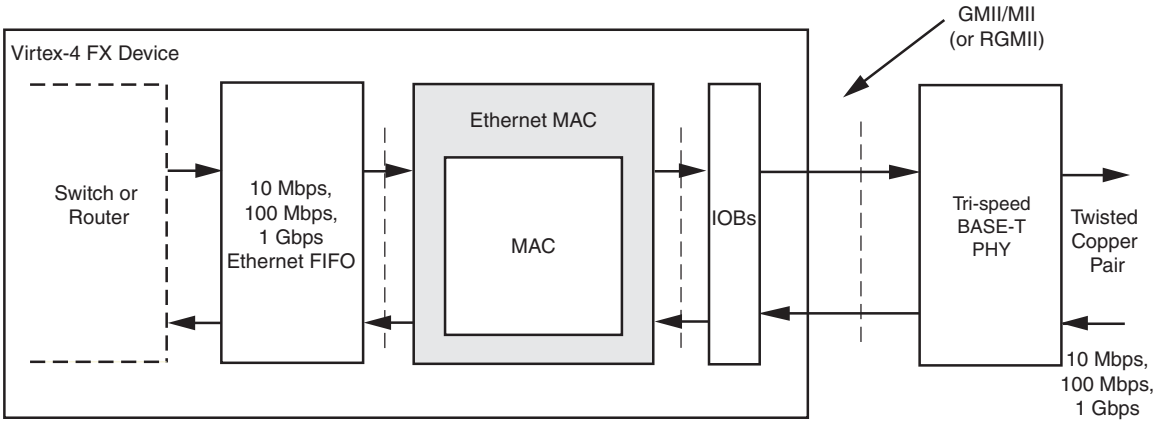

Figure 2: **Typical 1000BASE-T Application**

Ethernet 1000BASE-X Port

Figure 3 illustrates a typical application for a single Ethernet MAC. The PHY side of the MAC is connected to a RocketIO, which in turn is connected to an external off-the-shelf GBIC or SFP optical transceiver. The 1000BASE-X logic can be optionally provided by the Ethernet MAC, as displayed. 1000BASE-X functionality is demonstrated in the HDL examples provided with the example design.

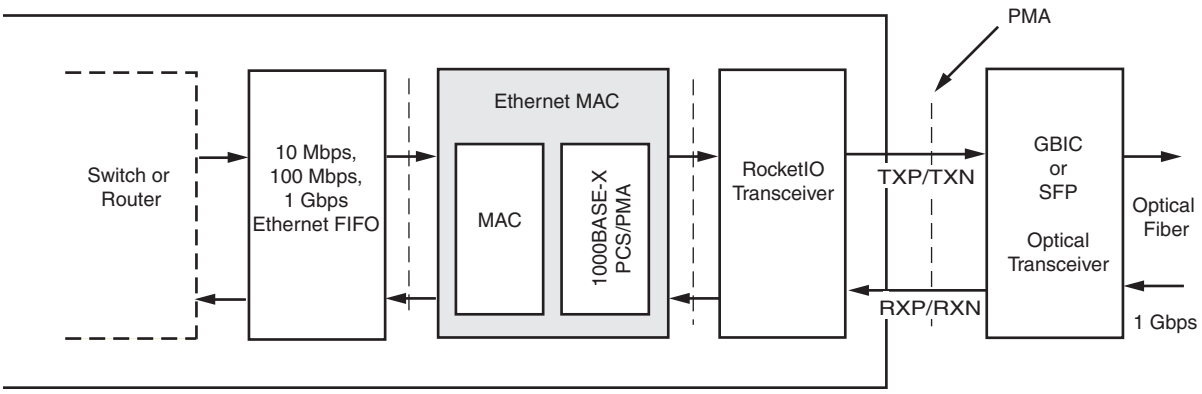

Figure 3: **Typical 1000BASE-X Application**

Example Design Overview

Figure 4 displays the major functional blocks of the Virtex-4 Tri-Mode Ethernet MAC example design. All illustrated components are provided in HDL with the exception of the Ethernet MAC itself.

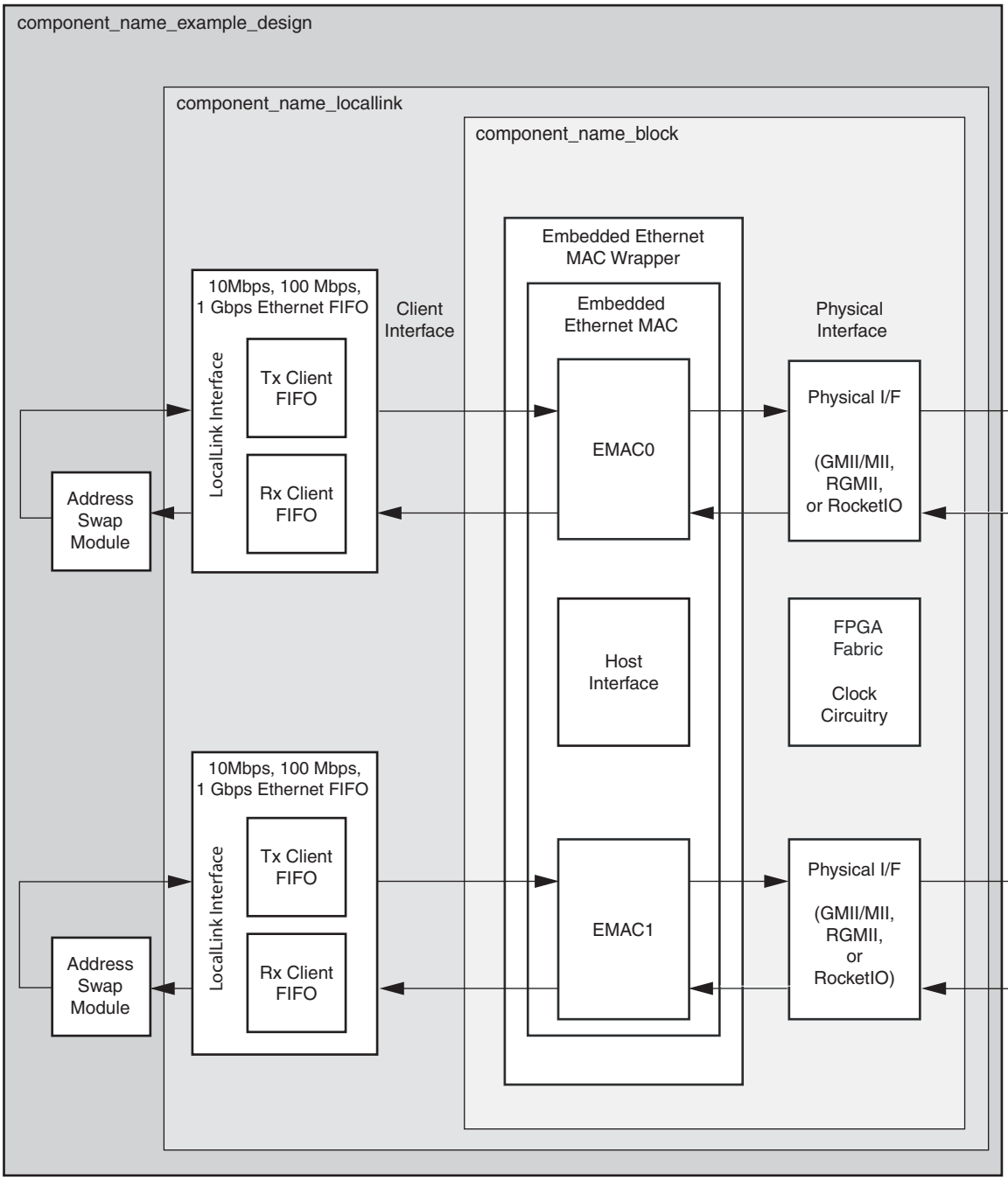

Figure 4: **Example Design**

Ethernet MAC Example Design

The example design is designed for quick adaptation and can be downloaded onto an FPGA to provide a real hardware test environment, and includes all the clock management logic required to operate the Ethernet MAC and its example design. DCMs, BUFGs, and so forth, are instantiated as required.

In the example design, the data is looped back at the client interface, enabling the Ethernet MAC to be quickly connected to a protocol tester—frames injected into the Ethernet MAC PHY Receive port are relayed back through the Ethernet MAC and out through the Ethernet MACs PHY Transmit port. Using this method, they are received back at the protocol tester.

The design includes an Address Swapping Module and a FIFO. Frames received by the Ethernet MAC are passed through the Receive side of the FIFO. Data from the Receive side of the FIFO is passed into the Address Swap Module and then on to the Transmit side of the FIFO using a LocalLink interface. The Transmit FIFO queues frames for transmission and connects directly to the client side Transmit interface of the Ethernet MAC.

Address Swap Module

The Address Swap Module switches the Destination Address and Source Address within the received MAC frame. Using this method, frames received from a link partner, for example a protocol tester, are relayed back to the correct Destination Address.

10 Mbps, 100 Mbps, 1 Gbps Ethernet FIFO

The 10 Mbps, 100 Mbps, 1 Gbps Ethernet FIFO is a wrapper file around the Receive and Transmit FIFO components. These components can be used in more complex client applications, as illustrated in Figures 2 and 3. To use the FIFOs, the component_name_locallink component can be instantiated in the user design.

Receive Client FIFO

The Receive (Rx) Client FIFO, a 4k-byte FIFO implemented in block RAMS, can be used for more complex client applications and can be connected directly to the Rx Client Interface of the Ethernet MAC. The Rx Client provides a LocalLink connection for the user.

- The FIFO operates at all Ethernet speeds supported by the Ethernet MAC.
- The FIFO drops all frames marked as bad from the Ethernet MAC so that only error-free frames are passed to the Ethernet client.

Transmit (Tx) Client FIFO

The Transmit (Tx) Client FIFO, a 4k-byte FIFO implemented in block RAMS, can used for more complex client applications and can be connected directly to the Tx Client Interface of the Ethernet MAC. The Tx Client FIFO provides a LocalLink connection for the user.

- The FIFO operates at all Ethernet speeds supported by the Ethernet MAC.
- The FIFO is capable of half-duplex re-transmission. For this reason, if a collision occurs on the medium, the Ethernet MAC indicates a collision on the Tx Client interface and the FIFO automatically re-queues the frame for re-transmission.

Ethernet MAC Wrapper

The Ethernet MAC wrapper file instantiates the full Ethernet MAC primitive. For one or both Ethernet MACs (EMAC0/EMAC1), the following applies:

- All unused input ports on the primitive are tied to the appropriate logic level; all unused output ports are left unconnected.
- Tie-off vectors are connected based on options selected in the CORE Generator.
- Only used ports are connected to the ports of the wrapper file.

This simplified wrapper should be used as the instantiation template for the Ethernet MAC in customer designs.

Physical I/F

An appropriate Physical Interface is provided for each selected EMAC0/EMAC1. This interface connects the physical interface of the Ethernet MAC block to the I/O of the FPGA. As required, the following components are provided:

- For GMII/MII, this component contains Input/Output block (IOB) buffers and IOB flip-flops.
- For RGMII, this component contains contain IOB buffers and IOB Double-Data Rate flip-flops.
- For 1000BASE-X PCS/PMA or SGMII, this component instantiates and connects RocketIO(s).

Device Utilization

The following sections provide approximate device-utilization figures for common configurations of the Ethernet MAC and its example design, and are separated into the following sections:

- 1 Gbps Only Operation
- Tri-speed Operation
- 100 Mbps or 10 Mbps Operation

Of interest is the utilization of clock resources, specifically the global clock usage (GCLKs), which should influence the type of interface selected. These clock resource figures do not consider any clock buffers, which may be required for the Host Interface.

1 Gbps Only Operation

Table 1 defines approximate utilization figures for common configurations of the Tri-Mode Ethernet MAC and its example design when supporting 1 Gbps only operation.

Table 1: **Device Utilization for 1 Gbps Operation**

Parameter Values		Device Resources					
Physical Interface	**Ethernet MAC Usage**	**Slices**	**LUTs**	**Registers**	**Block RAMs**	**GCLKs**	**DCMs**
GMII	Single EMAC	387	451	443	4	2[1]	0
	Both EMACs	766	901	876	8	3[1]	0
RGMII 1.3	Single EMAC	366	420	432	4	2[1]	0
	Both EMACs	721	839	854	8	3[1]	0

Table 1: **Device Utilization for 1 Gbps Operation *(Continued)***

Parameter Values		Device Resources					
RGMII 2.0	Single EMAC	366	420	432	4	2[1]	0
	Both EMACs	721	839	854	8	3[1]	0
SGMII	Single EMAC	591	659	731	5	2	0
	Both EMACs	1107	1233	1350	10	2	0
1000BASE-X (8-bit client)	Single EMAC	493	609	566	4	2	0
	Both EMACs	915	1135	1022	8	2	0
1000BASE-X (16-bit client)	Single EMAC	512	539	651	4	4	1
	Both EMACs	928	989	1177	8	4	1

1. These implementations use IDELAY elements, which require a 200MHz reference clock for the associated IDELAYCTRL. The reference clock is not accounted for as a GCLK.

Tri-speed Operation

Table 2 provides approximate utilization figures for common configurations of the Tri-Mode Ethernet MAC and its example design when operating at 10 Mbps, 100 Mbps, or 1 Gbps.

Table 2: **Device Utilization for Tri-speed Operation**

Parameter Values		Device Resources					
Physical Interface	**Ethernet MAC Usage**	**Slices**	**LUTs**	**Registers**	**Block RAMs**	**GCLKs**	**DCMs**
GMII/MII (standard clocking)	Single EMAC	394	456	446	4	4[1]	0
	Both EMACs	787	909	888	8	8[1]	0
GMII/MII (advanced clocking: full duplex mode only)	Single EMAC	383	395	473	4	2[1]	0
	Both EMACs	768	768	942	8	4[1]	0
RGMII 1.3	Single EMAC	366	420	432	4	4[1]	0
	Both EMACs	729	839	860	8	8[1]	0
RGMII 2.0	Single EMAC	366	420	432	4	4[1]	0
	Both EMACs	729	839	860	8	8[1]	0
SGMII	Single EMAC	591	659	731	5	3	0
	Both EMACs	1112	1233	1355	10	4	0

1. These implementations use IDELAY elements, which require a 200MHz reference clock for the associated IDELAYCTRL. The reference clock is not accounted for as a GCLK.

100 Mbps or 10 Mbps Operation

Table 3 provides approximate utilization figures for common configurations of the Tri-Mode Ethernet MAC and its example design when operating at 10 Mbps or 100 Mbps.

Note: For all other interfaces, see Tri-speed Operation.

Table 3: **Device Utilization for 10 Mbps, 100 Mbps Operation**

Parameter Values		Device Resources					
Physical Interface	**Ethernet MAC Usage**	**Slices**	**LUTs**	**Registers**	**Block RAMs**	**GCLKs**	**DCMs**
MII (standard clocking)	Single EMAC	389	452	441	4	4	0
	Both EMACs	774	903	878	8	8	0
MII (advanced clocking: clock enables)	Single EMAC	366	367	456	4	2	0
	Both EMACS	908	734	731	8	4	0

Ordering Information

The Tri-Mode Embedded Ethernet MAC Wrapper is provided to all licensed Xilinx ISE customers at no cost and can be generated using the Xilinx CORE Generator v9.1i or higher. For additional information about this and other Xilinx IP products, see the Xilinx IP Center.

Revision History

Date	Version	Revision
10/19/04	1.0	Initial Xilinx release.
2/25/05	2.0	Added support for SGMII and 1000BASE-X PCS/PMA physical interfaces.
7/26/05	3.1	Updated Features section to match version 3.1 of the wrapper.
1/18/06	4.1	Restructured document for new wrapper architecture.
7/13/06	4.2	Updated core to version 4.2; Xilinx tools 8.2i.
9/21/06	4.3	Updated core to version 4.3.
2/15/07	4.4	Updated core to version 4.4; ISE tools to 9.1i.