

# **Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC**

## ***User Guide***

UG368 (v1.3) March 1, 2011



**[www.BDTIC.com/XILINX](http://www.BDTIC.com/XILINX)**



Xilinx is disclosing this user guide, manual, release note, and/or specification (the "Documentation") to you solely for use in the development of designs to operate with Xilinx hardware devices. You may not reproduce, distribute, republish, download, display, post, or transmit the Documentation 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. Xilinx expressly disclaims any liability arising out of your use of the Documentation. Xilinx reserves the right, at its sole discretion, to change the Documentation without notice at any time. Xilinx assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU "AS-IS" WITH NO WARRANTY OF ANY KIND. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

©2009–2011 Xilinx, Inc. XILINX, the Xilinx logo, Virtex, Spartan, ISE, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. AMBA is a registered trademark of ARM in the EU and other countries. All other trademarks are the property of their respective owners.

## Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/24/09	1.0	Initial Xilinx release.
10/12/09	1.1	Updated the preface. Chapter 1: Updated <a href="#">Frame Transmission and Interframe Gap</a> . Chapter 2: Updated EMAC_RXFLOWCTRL_ENABLE and EMAC_TXFLOWCTRL_ENABLE descriptions in <a href="#">Table 2-17</a> . Chapter 3: Updated introductory paragraph in the <a href="#">Flow Control Block</a> section.
01/17/10	1.2	Updated <a href="#">Table 6-1</a> .
03/01/11	1.3	Updated <a href="#">Accessing the Ethernet MAC from the CORE Generator Tool</a> with note about core version 2.1. Updated description of EMAC_PAUSEADDR[47:0] in <a href="#">Table 2-17</a> . Added description of standard and alternative clock management to <a href="#">Transmitter Statistics Vector</a> and <a href="#">Receiver Statistics Vector</a> . Updated <a href="#">Table 4-14</a> . Updated 2.5 Gb/s serial transceiver settings for CLKFBOUT_MULT_F, CLKOUT0_DIVIDE_F, and CLKOUT1_DIVIDE in <a href="#">Table 6-1</a> . Added <a href="#">Use of Clock Correction Sequences</a> . Updated <a href="#">PHYEMACGTCLK</a> .

# Table of Contents

---

Revision History .....	2
<b>Preface: About This Guide</b>	
Guide Contents .....	9
Additional Documentation .....	10
Additional Support Resources .....	11
Acronyms .....	11
<b>Chapter 1: Introduction</b>	
Key Features .....	13
Typical Ethernet Application Overview .....	14
Ethernet Switch or Router .....	14
Ethernet Communications Port for an Embedded Processor .....	15
Ethernet Protocol Overview .....	16
Ethernet Sublayer Architecture .....	16
MAC and MAC CONTROL Sublayer .....	16
Physical Sublayers PCS, PMA, and PMD .....	17
Ethernet Data Format .....	17
Preamble .....	18
Start of Frame Delimiter .....	18
Destination Address .....	18
Source Address .....	18
Length/Type .....	18
Data .....	19
Pad .....	19
FCS .....	19
Frame Transmission and Interframe Gap .....	20
Half-Duplex Frame Transmission .....	20
Full-Duplex Frame Transmission .....	20
Using the Embedded Ethernet MAC .....	21
Accessing the Ethernet MAC from the CORE Generator Tool .....	21
Simulating the Ethernet MAC Using SecureIP Models .....	22
Model Considerations .....	22
<b>Chapter 2: Ethernet MAC Overview</b>	
Architecture Description .....	23
Ethernet MAC Configuration Options .....	25
Ethernet MAC Primitive .....	26
Ethernet MAC Signal Descriptions .....	28
Client-Side Signals .....	28
TX Client Signals .....	28
RX Client Signals .....	29
Flow Control Signals .....	30

Host Interface Signals . . . . .	31
Generic Host Bus Signals . . . . .	31
DCR Bus Signals . . . . .	32
Management Data Input/Output (MDIO) Signals . . . . .	32
MII/GMII/RGMII Physical Interface Signals . . . . .	32
Data and Control Signals . . . . .	33
Serial Transceiver Signals . . . . .	34
Global Clock and Reset Signals . . . . .	36
Global Clock Signal. . . . .	36
Reset Signal. . . . .	36
CLIENTEMACDCMLOCKED Signal. . . . .	36
<b>Ethernet MAC Attributes</b> . . . . .	36

## Chapter 3: Client Interface

<b>Transmit (TX) Client: 8-Bit Interface (without Clock Enables)</b> . . . . .	46
Normal Frame Transmission with 8-Bit Client Interface . . . . .	47
In-Band Parameter Encoding . . . . .	47
Padding . . . . .	47
Client-Supplied FCS Passing for Transmission . . . . .	47
Client Underrun . . . . .	48
Back-to-Back Transfers with 8-Bit Client Interface . . . . .	49
Virtual LAN (VLAN) Tagged Frames. . . . .	50
Maximum Permitted Frame Length/Jumbo Frames for Transmission . . . . .	50
Frame Collisions - Half-Duplex 10/100 Mb/s Operation Only . . . . .	51
IFG Adjustment . . . . .	52
<b>Transmit (TX) Client: 8-Bit Interface (with Clock Enables)</b> . . . . .	53
Normal Frame Transmission with 8-Bit Client Interface and Clock Enables . . . . .	53
<b>Transmit (TX) Client: 16-Bit Interface</b> . . . . .	54
Back-to-Back Transfers with 16-Bit Client Interface . . . . .	56
<b>Receive (RX) Client: 8-Bit Interface (without Clock Enables)</b> . . . . .	58
Normal Frame Reception . . . . .	58
Signal Timing of Good/Bad Frame Pulses . . . . .	59
Frame Reception with Errors . . . . .	59
Standard Conditions . . . . .	60
1000BASE-X/SGMII Specific Conditions . . . . .	60
Client-Supplied FCS Passing for Reception . . . . .	60
VLAN Tagged Frames. . . . .	61
Maximum Permitted Frame Length/Jumbo Frames for Reception . . . . .	61
Length/Type Field Error Checks. . . . .	61
Enabled. . . . .	61
Disabled . . . . .	62
Control Frame Length Checks . . . . .	62
<b>Receive (RX) Client: 8-Bit Interface (with Clock Enables)</b> . . . . .	63
<b>Receive (RX) Client: 16-Bit Interface</b> . . . . .	64
<b>Address Filtering</b> . . . . .	66
Address Filter Attributes . . . . .	66
Client RX Data/Control Interface . . . . .	67
<b>Flow Control Block</b> . . . . .	69
Requirement for Flow Control . . . . .	69
Flow Control Basics . . . . .	70
Transmitting a PAUSE Control Frame . . . . .	70

Receiving a PAUSE Control Frame . . . . .	72
Flow Control Implementation Example . . . . .	72
Method . . . . .	72
Operation . . . . .	73
<b>Statistics Vectors . . . . .</b>	<b>74</b>
Transmitter Statistics Vector . . . . .	74
Receiver Statistics Vector . . . . .	77
Statistics Registers/Counters . . . . .	80

## Chapter 4: Host/DCR Bus Interfaces

<b>Introduction to the Ethernet MAC Host Interface . . . . .</b>	<b>81</b>
<b>Ethernet MAC Register Descriptions . . . . .</b>	<b>83</b>
Configuration Registers . . . . .	84
Address Filter Registers . . . . .	89
<b>Using the Host Bus . . . . .</b>	<b>92</b>
Host Bus Clocking Requirements . . . . .	92
Reading and Writing MAC Configuration Registers . . . . .	92
Reading and Writing Address Filter Registers . . . . .	94
PCS/PMA Sublayer or External Device Access via MDIO . . . . .	94
<b>Using the DCR Bus . . . . .</b>	<b>96</b>
DCR Bus Clocking Requirements . . . . .	96
Device Control Registers . . . . .	96
Interfacing to a Processor . . . . .	101
DCR Interrupts . . . . .	101
Ethernet MAC Configuration and Address Filter Access . . . . .	103
Reading from an Ethernet MAC Configuration Register . . . . .	103
Writing to an Ethernet MAC Configuration Register . . . . .	103
Reading from the General Address Table Register of the Address Filter Block . . . . .	104
Writing to the General Address Table Register of the Address Filter Block . . . . .	104
PCS/PMA Sublayer or External Device Access via MDIO . . . . .	105
Reading the PHY Registers Using MDIO . . . . .	106
Writing to the PHY Registers Using MDIO . . . . .	107
Accessing FPGA Logic via Unused Host Bus Pins . . . . .	107

## Chapter 5: MDIO Interface

<b>Introduction to MDIO . . . . .</b>	<b>112</b>
MDIO Bus System . . . . .	112
MDIO Transactions . . . . .	113
Write Transaction . . . . .	113
Read Transaction . . . . .	113
MDIO Addressing . . . . .	114
Physical Address (PHYAD) . . . . .	114
Register Address (REGAD) . . . . .	114
<b>MDIO Implementation in the Ethernet MAC . . . . .</b>	<b>115</b>
Accessing MDIO through the Host Interface . . . . .	115
Accessing the Internal PCS/PMA Sublayer Management Registers . . . . .	116
Accessing the Management Registers of an External PHY using MDIO . . . . .	116
Accessing PCS/PMA Sublayer Management Registers using MDIO . . . . .	117
<b>1000BASE-X PCS/PMA Management Registers . . . . .</b>	<b>118</b>
<b>SGMII Management Registers . . . . .</b>	<b>125</b>

## Chapter 6: Physical Interface

<b>Introduction to the Physical Interfaces</b> .....	134
GMII / MII .....	134
RGMII .....	135
SGMII .....	135
1000BASE-X Sublayers .....	135
<b>Media Independent Interface (MII)</b> .....	135
MII Standard Clock Management .....	137
MII Clock Management Using Clock Enables .....	138
<b>Gigabit Media Independent Interface (GMII)</b> .....	139
GMII Clock Management for 1 Gb/s Only .....	140
GMII Standard Clock Management for Tri-Speed Operation .....	142
GMII Clock Management for Tri-Speed Operation Using Clock Enables .....	144
<b>Reduced Gigabit Media Independent Interface (RGMII)</b> .....	146
RGMII Clock Management for 1 Gb/s Only .....	147
RGMII Version 1.3 .....	147
RGMII Version 2.0 .....	149
RGMII Standard Clock Management for Tri-Speed Operation .....	150
RGMII Version 1.3 .....	150
RGMII Version 2.0 .....	152
RGMII Clock Management for Tri-Speed Operation Using Clock Enables .....	153
RGMII Version 1.3 .....	153
RGMII Version 2.0 .....	156
<b>1000BASE-X PCS/PMA</b> .....	158
Ethernet MAC PCS/PMA Sublayer .....	158
Introduction to the 1000BASE-X PCS/PMA Implementation .....	158
PCS Receive Engine and Synchronization .....	159
Auto-Negotiation .....	160
PCS Management Registers .....	160
Ethernet MAC to Serial Transceiver Connections .....	160
1000BASE-X PCS/PMA Clock Management .....	161
8-Bit Data Client .....	161
16-Bit Data Client .....	162
1000BASE-X Auto-Negotiation .....	164
Overview of Operation .....	164
1000BASE-X Auto-Negotiation Link Timer .....	166
Auto-Negotiation Interrupt .....	166
Use of Clock Correction Sequences .....	166
Loopback When Using the PCS/PMA Sublayer for 1000BASE-X .....	168
<b>Serial Gigabit Media Independent Interface (SGMII)</b> .....	169
Ethernet MAC PCS/PMA Sublayer .....	169
Introduction to the SGMII Implementation .....	170
PCS Transmit Engine .....	171
PCS Receive Engine and Synchronization .....	171
Auto-Negotiation .....	171
PCS Management Registers .....	171
SGMII RX Elastic Buffer .....	171
RX Elastic Buffer Implementations .....	172
Using the Serial Transceiver RX Elastic Buffer .....	176
Using the FPGA Logic Elastic Buffer .....	177
Serial Transceiver Logic Using the RX Elastic Buffer .....	177
Serial Transceiver Logic Using the RX Elastic Buffer in FPGA Logic .....	178

SGMII Clock Management .....	180
Tri-Speed Operation .....	180
1 Gb/s Only Operation .....	182
SGMII Auto-Negotiation .....	182
Overview of Operation .....	182
SGMII Auto-Negotiation Link Timer .....	183
Auto-Negotiation Interrupt .....	184
Loopback When Using the PCS/PMA Sublayer for SGMII .....	184
Switching Between SGMII and 1000BASE-X Standards .....	185
Performing a Mode Switch .....	186
Requirements and Considerations for Mode Switching .....	186

## Chapter 7: Interfacing to a Statistics Block

Using the Host Bus to Access Statistics Registers .....	187
Using the DCR Bus to Access Statistics Registers .....	188

## Appendix A: Pinout Requirements

Regional Clocking Requirements .....	191
--------------------------------------	-----

## Appendix B: Ethernet MAC Clocks

Ethernet MAC Internal Clock Logic Overview .....	193
Ethernet MAC Clock Generation .....	194
Ethernet MAC Input Clocks .....	195
Clock Connections to and from FPGA Logic .....	195
Standard Clocking Scheme .....	195
Transmitter Client Clock .....	196
Receiver Client Clock .....	196
Transmitter Physical Clock .....	197
Receiver Physical Clock .....	197
The Clock Enable Advanced Clocking Scheme .....	197
Clock Definitions and Frequencies .....	199
PHYEMACGTCLK .....	199
MII, 1 Gb/s GMII, 1 Gb/s RGMII, and Tri-Speed GMII with Clock Enables Modes ..	199
All Other Modes .....	199
PHYEMACMIITXCLK .....	199
MII or Tri-Speed GMII Modes .....	199
1000BASE-X PCS/PMA (16-Bit Data Client) Mode .....	200
All Other Modes .....	200
PHYEMACRXCLK .....	200
GMII, MII, and RGMII Modes .....	200
SGMII and 1000BASE-X PCS/PMA (8-Bit Data Client) Modes .....	200
1000BASE-X PCS/PMA (16-Bit Data Client) Mode .....	200
EMACPHYTXGMIIIMICLKOUT, PHYEMACTXGMIIIMICLKIN .....	200
MII, GMII, and RGMII Modes .....	200
SGMII and 1000BASE-X PCS/PMA Modes .....	201
EMACPHYTXCLK .....	201
EMACCLIENTTXCLIENTCLKOUT, CLIENTEMACTXCLIENTCLKIN .....	201
Clock Enables Not in Use .....	201
Clock Enables in Use .....	201
EMACCLIENTRXCLIENTCLKOUT, CLIENTEMACRXCLIENTCLKIN .....	202

Clock Enables Not in Use .....	202
Clock Enables in Use .....	202

## Appendix C: Virtex-5 to Virtex-6 FPGA Enhancements

<b>New Features</b> .....	203
Dynamic Switching Between SGMII and 1000BASE-X Standards .....	203
Programmable Auto-Negotiation Link Timer .....	203
Up to 2.5 Gb/s Overclocked Operation .....	203
Control Frame Length Check Disable .....	204
Ignore MDIO Broadcast Address .....	204
Address Match Statistics Bit .....	204
Address Filter Changes .....	204
Hardened Shim .....	204
<b>Port Map Changes</b> .....	204
<b>Attribute Changes</b> .....	205
<b>Use Model Changes</b> .....	205

## Appendix D: Differences between the Embedded Tri-Mode Ethernet MAC and the Soft IP Cores

Features Exclusive to the Embedded Tri-Mode Ethernet MAC .....	207
Features Exclusive to Soft IP Cores .....	207



# About This Guide

---

This user guide is a description of the Virtex®-6 FPGA Embedded Tri-Mode Ethernet MAC. Complete and up-to-date documentation of the Virtex-6 family of FPGAs is available on the Xilinx® website at:

<http://www.xilinx.com/support/documentation/virtex-6.htm>.

## Guide Contents

This user guide contains the following chapters:

- [Chapter 1, Introduction](#), introduces the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC and describes how to get started using the Ethernet MAC.
- [Chapter 2, Ethernet MAC Overview](#), describes the architecture of the Ethernet MAC, defines its signal interface, and gives an overview of its possible configurations.
- [Chapter 3, Client Interface](#), provides details of the client interface protocol and the Ethernet functionality associated with the client interface.
- [Chapter 4, Host/DCR Bus Interfaces](#), describes how to access registers in the Ethernet MAC using either the generic host bus or the DCR bus. The Ethernet MAC register descriptions are also found in this chapter.
- [Chapter 5, MDIO Interface](#), describes the MDIO interface protocol and the MDIO implementation in the Ethernet MAC. The SGMII and 1000BASE-X PCS/PMA management register descriptions are also found in this chapter.
- [Chapter 6, Physical Interface](#), describes all of the possible configurations for the physical interface and includes a description of optimized clocking schemes that can be used with the Ethernet MAC.
- [Chapter 7, Interfacing to a Statistics Block](#), provides details on how to use the Ethernet MAC with an FPGA logic-based statistics block.
- [Appendix A, Pinout Requirements](#), lists pinout requirements as well as recommendations to improve design timing when using the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC.
- [Appendix B, Ethernet MAC Clocks](#), gives an overview of the internal clock logic of the Ethernet MAC and summarizes the main clocking schemes. Definitions of each clock and its frequency, in all possible Ethernet MAC configurations, are also provided.
- [Appendix C, Virtex-5 to Virtex-6 FPGA Enhancements](#), documents some key enhancements and a few minor modifications.
- [Appendix D, Differences between the Embedded Tri-Mode Ethernet MAC and the Soft IP Cores](#), explains the differences between soft IP cores and the Tri-Mode Ethernet MAC by listing the features exclusive to both.

## Additional Documentation

The following documents are also available for download at: <http://www.xilinx.com/6>.

- **Virtex-6 Family Overview**  
The features and product selection of the Virtex-6 family are outlined in this overview.
- **Virtex-6 FPGA Data Sheet: DC and Switching Characteristics**  
This data sheet contains the DC and Switching Characteristic specifications for the Virtex-6 family.
- **Virtex-6 FPGA Packaging and Pinout Specifications**  
This specification includes the tables for device/package combinations and maximum I/Os, pin definitions, pinout tables, pinout diagrams, mechanical drawings, and thermal specifications.
- **Virtex-6 FPGA Configuration User Guide**  
This all-encompassing configuration guide includes chapters on configuration interfaces (serial and SelectMAP), bitstream encryption, boundary-scan and JTAG configuration, reconfiguration techniques, and readback through the SelectMAP and JTAG interfaces.
- **Virtex-6 FPGA SelectIO Resources User Guide**  
This guide describes the SelectIO™ resources available in all Virtex-6 devices.
- **Virtex-6 FPGA Clocking Resources User Guide**  
This guide describes the clocking resources available in all Virtex-6 devices, including the MMCM and PLLs.
- **Virtex-6 FPGA Memory Resources User Guide**  
The functionality of the block RAM and FIFO are described in this user guide.
- **Virtex-6 FPGA Configurable Logic Blocks User Guide**  
This guide describes the capabilities of the configurable logic blocks (CLBs) available in all Virtex-6 devices.
- **Virtex-6 FPGA GTH Transceivers User Guide**  
This guide describes the GTH transceivers available in all Virtex-6 HXT FPGAs except the XC6VHX250T and the XC6VHX380T in the FF1154 package.
- **Virtex-6 FPGA GTX Transceivers User Guide**  
This guide describes the GTX transceivers available in all Virtex-6 FPGAs except the XC6VLX760.
- **Virtex-6 FPGA DSP48E1 Slice User Guide**  
This guide describes the DSP48E1 slice available in all Virtex-6 FPGAs.
- **Virtex-6 FPGA System Monitor User Guide**  
The System Monitor functionality available in all Virtex-6 devices is outlined in this guide.
- **Virtex-6 FPGA PCB Designer's Guide**  
This guide provides information on PCB design for Virtex-6 devices, with a focus on strategies for making design decisions at the PCB and interface level.

The following documents are also available for download at:

[http://www.xilinx.com/support/documentation/ip\\_documentation.htm](http://www.xilinx.com/support/documentation/ip_documentation.htm).

- Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper Getting Started Guide  
Introduction to the benefits, usage guidelines, and debugging guidelines of the CORE Generator™ tool Ethernet MAC wrapper.
- Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper Data Sheet  
The data sheet for the CORE Generator tool Ethernet MAC wrapper.

## Additional Support Resources

To find additional documentation, see the Xilinx website at:

<http://www.xilinx.com/support/documentation/index.htm>.

To search the Answer Database of silicon, software, and IP questions and answers, or to create a technical support WebCase, see the Xilinx website at:

<http://www.xilinx.com/support>.

## Acronyms

Following is the list of acronyms and abbreviations that are used throughout this User Guide.

Acronym	Definition
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DA	Destination Address (MAC frame)
DCR	Device Control Register
DMA	Direct Memory Access
EDK	Embedded Development Kit (Xilinx software)
FCS	Frame Check Sequence (MAC frame)
FIFO	First In, First Out memory
FPGA	Field Programmable Gate Array
GBIC	Gigabit Interface Converter (optical transceiver)
Gb/s	Gigabits per second
GMII	Gigabit Media Independent Interface
GTX	High-speed serial transceivers offered in the Virtex-6 LXT and SXT platforms
IFG	Interframe Gap (MAC frame)
IOB	Input/Output Block
ISE	Xilinx Integrated Software Environment
L/T	Length/Type field (MAC frame)

Acronym	Definition
LAN	Local Area Network
MAC	Media Access Controller
Mb/s	Megabits per second
MDIO	Management Data Input/Output
MII	Media Independent Interface
MMCM	Mixed-Mode Clock Manager
OP	Operation code (read or write) for MDIO frame (sometimes called OPCODE in text)
OSI	Open Systems Interconnection
PCB	Printed Circuit Board
PCS	Physical Coding Sublayer
PHY	Physical Layer The term refers to all physical sublayers (PCS, PMA, PMD). Often applied to a device, e.g., BASE-T PHY: an external chip which can connect to the Ethernet MAC to perform this physical standard
PHYAD	Physical Address (MDIO Frame)
PLB	Processor Local Bus
PMA	Physical Medium Attachment
PMD	Physical Medium Dependent
PRE	Preamble (MDIO Frame)
REGAD	Register Address (MDIO Frame)
RGMII	Reduced Gigabit Media Independent Interface
RX	Receiver
SA	Source Address (MAC frame)
SFD	Start of Frame Delimiter (MAC frame)
SFP	Small Form-factor Pluggable (optical transceiver)
SGMII	Serial Gigabit Media Independent Interface
ST	Start Code (MDIO Frame) or Start of Frame
Stats	Abbreviation of Statistics
TA	Turnaround (MDIO Frame)
TX	Transmitter
VLAN	Virtual Local Area Network

# Introduction

---

This chapter introduces the Virtex®-6 FPGA Embedded Tri-Mode Ethernet MAC and provides an overview of typical Ethernet applications and the Ethernet protocol. It also provides guidance on how to successfully incorporate the Virtex-6 FPGA Ethernet Media Access Controller (MAC) into a larger design using Xilinx® tools.

The Virtex-6 device family contains multiple embedded Ethernet MACs that are independently configurable to meet all common Ethernet system connectivity needs. Table 1 in the *Virtex-6 Family Overview* lists the device type versus the number of available Ethernet MACs.

This chapter contains the following sections:

- [Key Features](#)
- [Typical Ethernet Application Overview](#)
- [Ethernet Protocol Overview](#)
- [Using the Embedded Ethernet MAC](#)

## Key Features

The key features of the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC are:

- Fully integrated 10/100/1000 Mb/s Ethernet MAC
- Designed to the IEEE Std 802.3-2005 specification
- Configurable full-duplex operation in 10/100/1000 Mb/s
- Configurable half-duplex operation in 10/100 Mb/s
- Media Independent Interface (MII) and Gigabit Media Independent Interface (GMII) when operating at 2.5V only, and Reduced Gigabit Media Independent Interface (RGMII)
- 1000BASE-X Physical Coding Sublayer (PCS) and a Physical Medium Attachment (PMA) sublayer included for use with the Virtex-6 serial transceivers to provide a complete on-chip 1000BASE-X implementation
- Support for 1000BASE-X overclocking up to 2500 Mb/s
- Serial Gigabit Media Independent Interface (SGMII) supported through the serial transceivers' interfaces to external copper PHY layer for full-duplex operation
- Management Data Input/Output (MDIO) interface to manage objects in the physical layer
- User-accessible raw statistic vector outputs
- Support for VLAN frames
- Configurable interframe gap (IFG) adjustment in full-duplex operation

- Configurable in-band Frame Check Sequence (FCS) field passing on both transmit and receive paths
- Auto padding on transmit and stripping on receive paths
- Configured and monitored through a host interface
- Hardware-selectable Device Control Register (DCR) bus or generic host bus interface
- Configurable flow control through Ethernet MAC Control PAUSE frames; symmetrically or asymmetrically enabled
- Configurable support for jumbo frames of any length
- Configurable receive address filter for unicast, general, and broadcast addresses

## Typical Ethernet Application Overview

Typical applications for the Ethernet MAC include:

- [Ethernet Switch or Router](#)
- [Ethernet Communications Port for an Embedded Processor](#)

### Ethernet Switch or Router

[Figure 1-1](#) illustrates a typical application for a single Ethernet MAC. The PHY side of the core is connected to an off-the-shelf Ethernet PHY device, which performs the BASE-T standard at 1 Gb/s, 100 Mb/s, and 10 Mb/s speeds. The PHY device can be connected using any of the following supported interfaces: GMII/MII, RGMII, or SGMII.

The client side of the Embedded Ethernet MAC is connected to a FIFO to complete a single Ethernet port. This port is connected to a Switch or Routing matrix, which can contain several ports.

For this application, the recommended place to start is in [Accessing the Ethernet MAC from the CORE Generator™ tool, page 21](#). The CORE Generator™ tool provides an example design for the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC for any of the supported physical interfaces. A FIFO example is also generated, which can be used as the FIFO in the illustration, for a typical application.

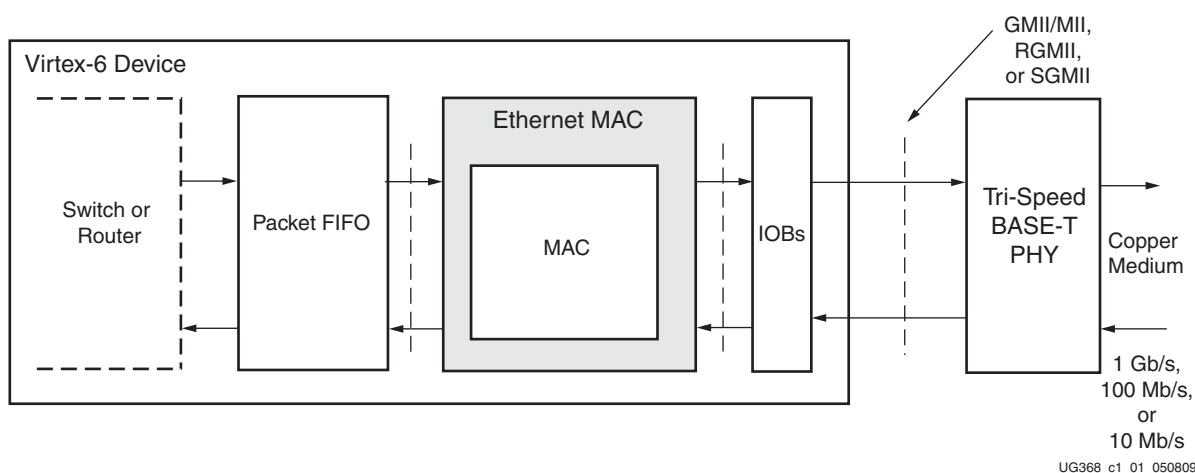


Figure 1-1: Typical Application: Ethernet Switch or Router

## Ethernet Communications Port for an Embedded Processor

Figure 1-2 illustrates a typical application for a single Ethernet MAC. The PHY side of the core is connected to an off-the-shelf Ethernet PHY device, which performs the BASE-T standard at 1 Gb/s, 100 Mb/s, and 10 Mb/s speeds. The PHY device can be connected using any of the following supported interfaces: GMII/MII, RGMII, or SGMII.

A soft core is provided as part of the Xilinx® Platform Studio (XPS), Embedded Development Kit (EDK) IP portfolio to connect the client interface of the Embedded Ethernet MAC to the DMA port of a processor. [DS537](#), *XPSLL TEMAC Data Sheet*, describes the XPS\_LL\_TEMAC, which can be instantiated for an intended processor application.

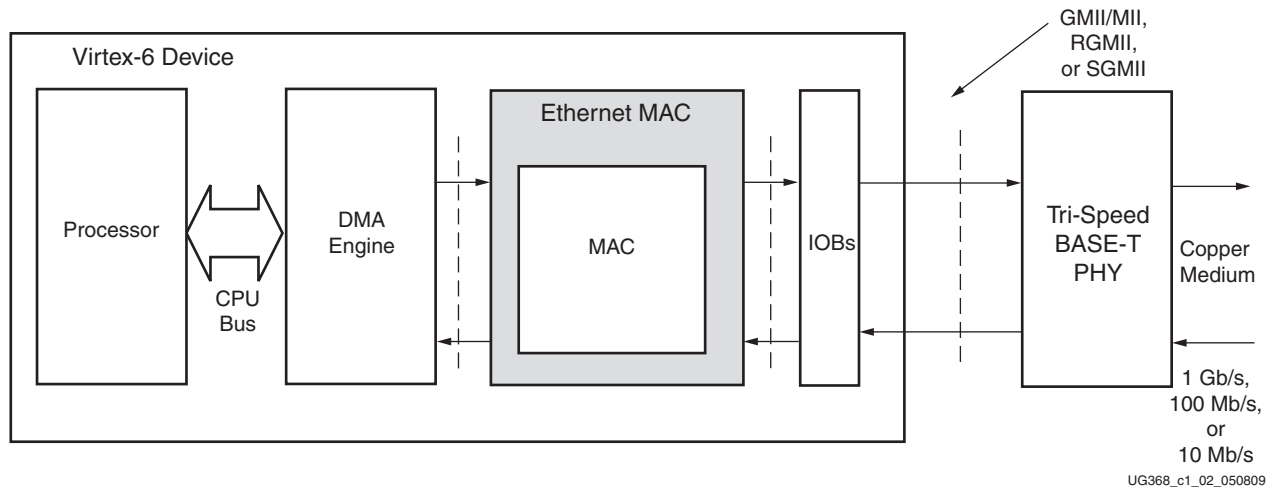


Figure 1-2: Typical Application: Ethernet Communications Port for Embedded Processor

## Ethernet Protocol Overview

This section gives an overview of where the Ethernet MAC fits into an Ethernet system and provides a description of some basic Ethernet terminology.

### Ethernet Sublayer Architecture

Figure 1-3 illustrates the relationship between the OSI reference model and the Ethernet MAC, as defined in IEEE Std 802.3-2005. The grayed-in layers show the functionality that the Ethernet MAC handles. Figure 1-3 also shows where the supported physical interfaces fit into the architecture.

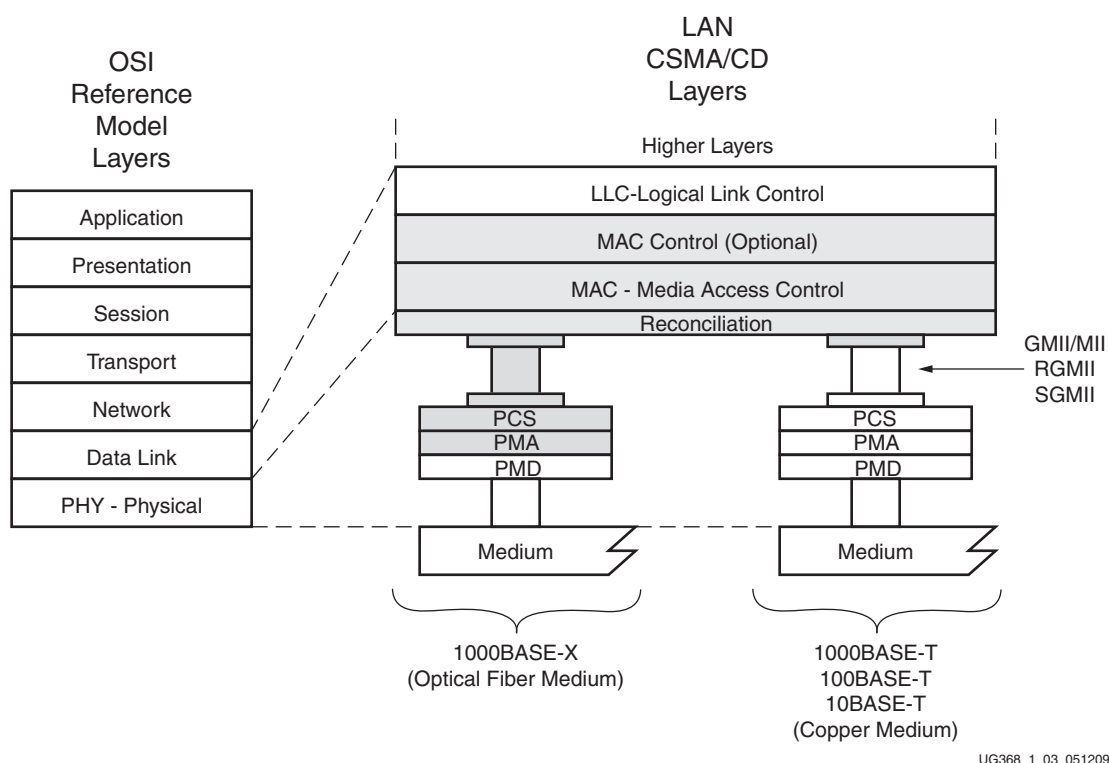


Figure 1-3: IEEE Std 802.3-2005 Ethernet Model

### MAC and MAC CONTROL Sublayer

The Ethernet MAC is defined in IEEE Std 802.3-2005, clauses 2, 3, and 4. A MAC is responsible for the Ethernet framing protocols described in Ethernet Data Format and error detection of these frames. The MAC is independent of and can connect to any type of physical layer device.

The MAC CONTROL sublayer is defined in IEEE Std 802.3-2005, clause 31. This provides real-time flow control manipulation of the MAC sublayer.

Both the MAC CONTROL and MAC sublayers are provided by the Ethernet MAC in all modes of operation.



## Physical Sublayers 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:

- **BASE-T** PHYs provide a link between the MAC and copper mediums. This functionality is not offered within the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC. However, external BASE-T PHY devices are readily available on the market. These can connect to the Ethernet MAC, using GMII/MII, RGMII, or SGMII interfaces.
- **BASE-X** PHYs provide a link between the MAC and (usually) fiber optic mediums. The Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC is capable of supporting the 1 Gb/s BASE-X standard; 1000BASE-X PCS and PMA sublayers can be offered by connecting the Ethernet MAC to a serial transceiver. An optical transceiver can be directly connected to the serial transceiver to complete the PMD sublayer functionality.

## Ethernet Data Format

Ethernet data is encapsulated in frames, as shown in Figure 1-4, for standard Ethernet frames. The fields in the frame are transmitted from left to right. The bytes within the frame are transmitted from left to right (from least significant bit to most significant bit unless specified otherwise). The Ethernet MAC can handle jumbo Ethernet frames where the data field can be much larger than 1500 bytes.

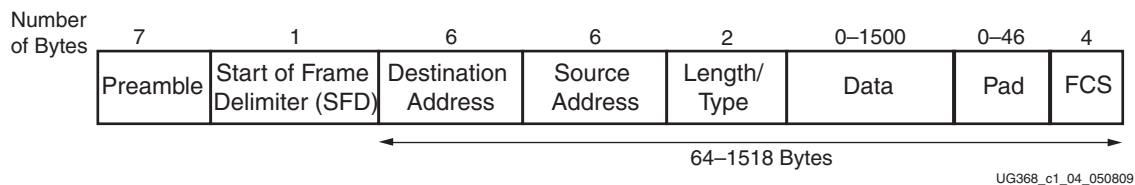


Figure 1-4: Standard Ethernet Frame Format

The Ethernet MAC can also accept VLAN frames. The VLAN frame format is shown in Figure 1-5. If the frame is a VLAN type frame, the Ethernet MAC accepts four additional bytes.

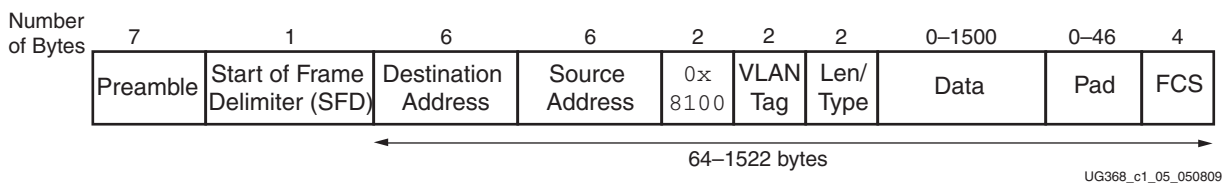


Figure 1-5: Ethernet VLAN Frame Format

Ethernet PAUSE/flow control frames can be transmitted and received by the Ethernet MAC. Figure 3-29, page 71 shows how a PAUSE/flow control frame differs from the standard Ethernet frame format.

The following subsections describe the individual fields of an Ethernet frame and some basic functionality of the Ethernet MAC.

## Preamble

For transmission, this field is automatically inserted by the Ethernet MAC. The preamble field was historically used for synchronization and contains seven bytes with the pattern 0x55, transmitted from left to right. For reception, this field is always stripped from the incoming frame, before the data is passed to the client. The Ethernet MAC can receive Ethernet frames, even if the preamble does not exist, as long as a valid start of frame delimiter is available.

## Start of Frame Delimiter

The start of frame delimiter field marks the start of the frame and must contain the pattern 0xD5.

For transmission on the physical interface, this field is automatically inserted by the Ethernet MAC. For reception, this field is always stripped from the incoming frame before the data is passed to the client.

## Destination Address

The least significant bit of the destination address determines if the address is an individual/unicast (0) or group/multicast (1) address. Multicast addresses are used to group logically related stations. The broadcast address (destination address field is all 1s) is a multicast address that addresses all stations on the LAN. The Ethernet MAC supports transmission and reception of unicast, multicast, and broadcast packets.

This field is the first field of the Ethernet frame that is always provided in the packet data for transmissions and is always retained in the receive packet data.

## Source Address

For transmission, the source address of the Ethernet frame should be provided by the client because it is unmodified by the Ethernet MAC. The unicast address for the Ethernet MAC is used as the source address when the Ethernet MAC creates PAUSE control frames. The source address field is always retained in the receive packet data.

## Length/Type

The value of this field determines if it is interpreted as a length or a type field, as defined by IEEE Std 802.3-2005. A value of 1536 decimal or greater is interpreted by the Ethernet MAC as a type field.

When used as a length field, the value in this field represents the number of bytes in the following data field. This value does not include any bytes that can be inserted in the pad field following the data field.

A length/type field value of 0x8100 indicates that the frame is a VLAN frame, and a value of 0x8808 indicates a PAUSE MAC control frame.

For transmission, the Ethernet MAC does not perform any processing of the length/type field.

For reception, if this field is a length field, the Ethernet MAC receive engine interprets this value and removes any padding in the pad field (if necessary). If the field is a length field and length/type checking is enabled, the Ethernet MAC compares the length against the actual data field length and flags an error if a mismatch occurs. If the field is a type field, the Ethernet MAC ignores the value and passes it along with the packet data with no further processing. The length/type field is always retained in the receive packet data.

## Data

The data field can vary from 0 to 1,500 bytes in length for a normal frame. The Ethernet MAC can handle jumbo frames of any length.

This field is always provided in the packet data for transmissions and is always retained in the receive packet data.

## Pad

The pad field can vary from 0 to 46 bytes in length. This field is used to ensure that the frame length is at least 64 bytes in length (the preamble and SFD fields are not considered part of the frame for this calculation), which is required for successful CSMA/CD operation. The values in this field are used in the frame check sequence calculation but are not included in the length field value, if it is used. The length of this field and the data field combined must be at least 46 bytes. If the data field contains 0 bytes, the pad field is 46 bytes. If the data field is 46 bytes or more, the pad field has 0 bytes.

For transmission, this field can be inserted automatically by the Ethernet MAC or can be supplied by the client. If the pad field is inserted by the Ethernet MAC, the FCS field is calculated and inserted by the Ethernet MAC. If the pad field is supplied by the client, the FCS can be either inserted by the Ethernet MAC or provided by the client, as indicated by a configuration register bit.

For reception, if the length/type field has a length interpretation, any pad field in the incoming frame is not be passed to the client, unless the Ethernet MAC is configured to pass the FCS field on to the client.

## FCS

The value of the FCS field is calculated over the destination address, source address, length/type, data, and pad fields using a 32-bit Cyclic Redundancy Check (CRC), as defined in IEEE Std 802.3-2005 para. 3.2.8:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + x^0$$

The CRC bits are placed in the FCS field with the  $x^{31}$  term in the left-most bit of the first byte, and the  $x^0$  term is the right-most bit of the last byte (i.e., the bits of the CRC are transmitted in the order  $x^{31}, x^{30}, \dots, x^1, x^0$ ).

For transmission, this field can be either inserted automatically by the Ethernet MAC or supplied by the client, as indicated by a configuration register bit.

For reception, the incoming FCS value is verified on every frame. If an incorrect FCS value is received, the Ethernet MAC indicates to the client that it has received a bad frame. The FCS field can either be passed on to the client or be dropped by the Ethernet MAC, as indicated by a configuration register bit.

## Frame Transmission and Interframe Gap

Frames are transmitted over the Ethernet medium with an interframe gap, as specified by the IEEE Std 802.3-2005. For full-duplex systems, the minimum IFG is 96 bit times (9.6  $\mu$ s for 10 Mb/s, 0.96  $\mu$ s for 100 Mb/s, and 96 ns for 1 Gb/s), as defined by IEEE Std 802.3-2005. For half-duplex systems, the minimum IFG is 208 bit times when using the RGMII physical interface, and 144 bit times when using the MII physical interface. The defined IFG is a minimum and can be increased with a resulting decrease in throughput.

The process for frame transmission is different for half-duplex and full-duplex systems.

### Half-Duplex Frame Transmission

In a half-duplex system, the CSMA/CD media access method defines how two or more stations share a common medium.

1. Even when it has nothing to transmit, the Ethernet MAC monitors the Ethernet medium for traffic by watching the carrier sense signal (CRS) from the external PHY. Whenever the medium is busy (CRS = 1), the Ethernet MAC defers to the passing frame by delaying any pending transmission of its own.
2. After the last bit of the passing frame (when the carrier sense signal changes from TRUE to FALSE), the Ethernet MAC starts the timing of the interframe gap.
3. The Ethernet MAC resets the interframe gap timer if the carrier sense becomes TRUE during the period defined by "interframe gap part 1 (IFG1)." IEEE Std 802.3-2005 states that this should be the first 2/3 of the interframe gap timing interval (64 bit times) but can be shorter and as small as zero. The purpose of this option is to support a possible brief failure of the carrier sense signal during a collision condition and is described in paragraph 4.2.3.2.1 of the IEEE standard.
4. The Ethernet MAC does not reset the interframe gap timer if carrier sense becomes TRUE during the period defined by "interframe gap part 2 (IFG2)" to ensure fair access to the bus. IEEE Std 802.3-2005 states that this should be the last 1/3 of the interframe gap timing interval.

If, after initiating a transmission, the message collides with the message of another station (COL = 1), then each transmitting station intentionally continues to transmit (jam) for an additional predefined period (32 bit times for 10/100 Mb/s) to ensure propagation of the collision throughout the system. The station remains silent for a random amount of time (backoff) before attempting to transmit again.

A station can experience a collision during the beginning of its transmission (the collision window) before its transmission has had time to propagate to all stations on the bus. After the collision window has passed, a transmitting station has acquired the bus. Subsequent collisions (late collisions) are avoided because all other (properly functioning) stations are assumed to have detected the transmission and are deferring to it.

### Full-Duplex Frame Transmission

In a full-duplex system, there is a point-to-point dedicated connection between two Ethernet devices, capable of simultaneous transmit and receive with no possibility of collisions. The Ethernet MAC does not use the carrier sense signal from the external PHY because the medium is not shared, and the Ethernet MAC only needs to monitor its own transmissions. After the last bit of an Ethernet MAC frame transmission, the Ethernet MAC starts the interframe gap timer and defers transmissions until it reaches either 96 bit times, or when IFG Adjustment is enabled, the greater of 96 bit times and the value presented on CLIENTEMACTXIFGDELAY.

## Using the Embedded Ethernet MAC

This section describes how the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC can be easily incorporated into customer designs using Xilinx software:

- [Accessing the Ethernet MAC from the CORE Generator Tool](#): The CORE Generator™ tool provides wrapper files to help instantiate and configure the Ethernet MAC.
- [Simulating the Ethernet MAC Using SecureIP Models](#): SecureIP models are provided with the ISE® software to allow the Ethernet MAC to be simulated both functionally and with timing information.

This section provides more information on how to use these Xilinx software tools to access and configure the Ethernet MAC.

### Accessing the Ethernet MAC from the CORE Generator Tool

Generating the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC wrapper files greatly simplifies the use of the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC. The Ethernet MAC is highly configurable, and not all pins/interfaces are required for every configuration. The CORE Generator tool allows the configuration of the Ethernet MAC to be selected using a GUI and generates HDL wrapper files for the configuration. These wrapper files hide much of the complexity of the Ethernet MAC by only bringing out interface signals for the selected configuration.

**Note:** Beginning with core version 2.1, the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC wrapper provides AMBA® AXI4 interfaces, a modified register space, and enhanced functionality implemented in FPGA logic. Users wishing to take advantage of these new features should refer to the documentation provided with the core. This document is most appropriate for users of the embedded Ethernet MAC block itself, and Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC wrapper core versions below 2.1.

Accessing the Ethernet MAC from the CORE Generator tool provides the following features:

- Sets the values of the Ethernet MAC attributes based on user options
- Provides user-configurable Ethernet MAC physical interfaces
  - Supports MII, GMII, RGMII v1.3, RGMII v2.0, SGMII, and 1000BASE-X PCS/PMA interfaces
  - Provides off-chip connections for physical interfaces by instantiating serial transceivers, and logic as required, for the selected physical interfaces
- Provides an optimized clocking scheme for the selected physical interface and instantiates the required clock buffers, Mixed-Mode Clock Managers (MMCMs), etc.
- Provides a simple FIFO-loopback example design, which is connected to the MAC client interfaces
- Provides a simple demonstration testbench based on the selected configuration
- Provides a state machine to facilitate optional switching between SGMII and 1000BASE-X physical interface standards
- Generates VHDL or Verilog wrapper files

For further details, refer to *Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper Data Sheet* and the *Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC Wrapper Getting Started Guide*.

## Simulating the Ethernet MAC Using SecureIP Models

SecureIP models are encrypted versions of the actual HDL code that allow the user to simulate the functionality of the design without the overhead of simulating RTL. A Verilog LRM-IEEE 1364-2005 encryption-compliant simulator is required to use SecureIP.

The SecureIP model of the Ethernet MAC is installed with the Xilinx tools and can be precompiled into UniSim and SimPrim libraries. These libraries are used for functional and timing simulations, respectively. In addition, VHDL and Verilog wrappers are generated by the CORE Generator tool in the ISE software, as well as the scripts to simulate the Secure IP model.

For further help using the Ethernet MAC SecureIP model, see the documentation supplied with ISE software, especially the *Synthesis and Simulation Design Guide* at:

[http://www.xilinx.com/support/software\\_manuels.htm](http://www.xilinx.com/support/software_manuels.htm).

### Model Considerations

If the Ethernet MAC is configured in either SGMII or 1000BASE-X PCS/PMA mode, simulation times are much longer than when using other physical interfaces due to the auto-negotiation sequence. It is recommended that Auto Negotiation is disabled (by setting the EMAC\_PHYINITAUTONEG\_ENABLE attribute to FALSE or by disabling auto-negotiation by writing through the host interface) to allow frame transmission to occur within a reasonable period of simulation time. See [Ethernet MAC Attributes, page 36](#). If simulating Auto Negotiation, then simulation time can be reduced by setting the programmable link timer to a small value via the EMAC\_LINKINTERVAL[8:0] attribute or the Ethernet MAC Mode Configuration Register.

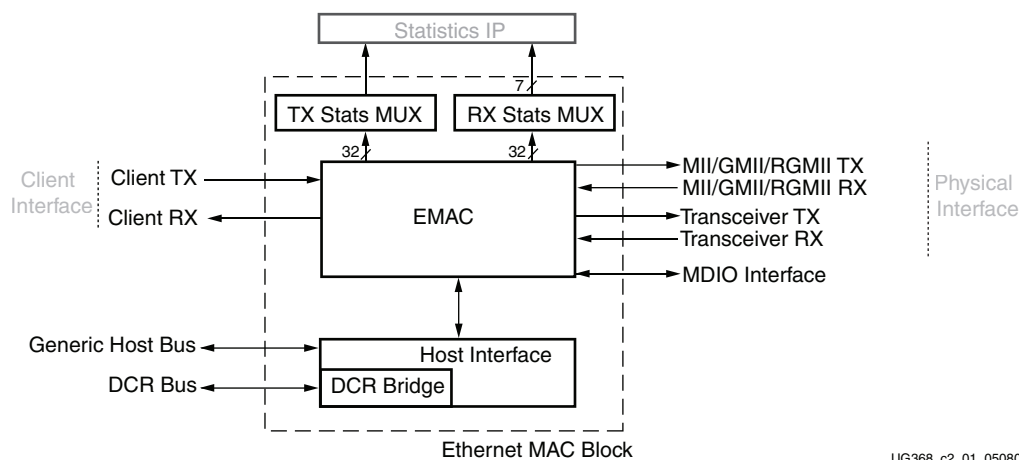
## Ethernet MAC Overview

This chapter describes the architecture of the Virtex®-6 FPGA Embedded Tri-Mode Ethernet Media Access Controller (MAC) block. It contains the following sections:

- [Architecture Description](#)
- [Ethernet MAC Configuration Options](#)
- [Ethernet MAC Primitive](#)
- [Ethernet MAC Signal Descriptions](#)
- [Ethernet MAC Attributes](#)

### Architecture Description

Figure 2-1 shows a block diagram of the Ethernet MAC block.



UG368\_c2\_01\_050809

Figure 2-1: Embedded Ethernet MAC Block

The host interface of the Ethernet MAC provides access to the configuration registers, illustrated in Figure 2-1. The host interface can be accessed using either the generic host bus or the DCR bus through the DCR bridge.

The physical interface can be configured as either MII, GMII, RGMII, SGMII, or 1000BASE-X. Figure 2-1 shows all possible data ports for the physical interface; however, only one set of these TX and RX interfaces is active, based on the selected physical interface configuration.

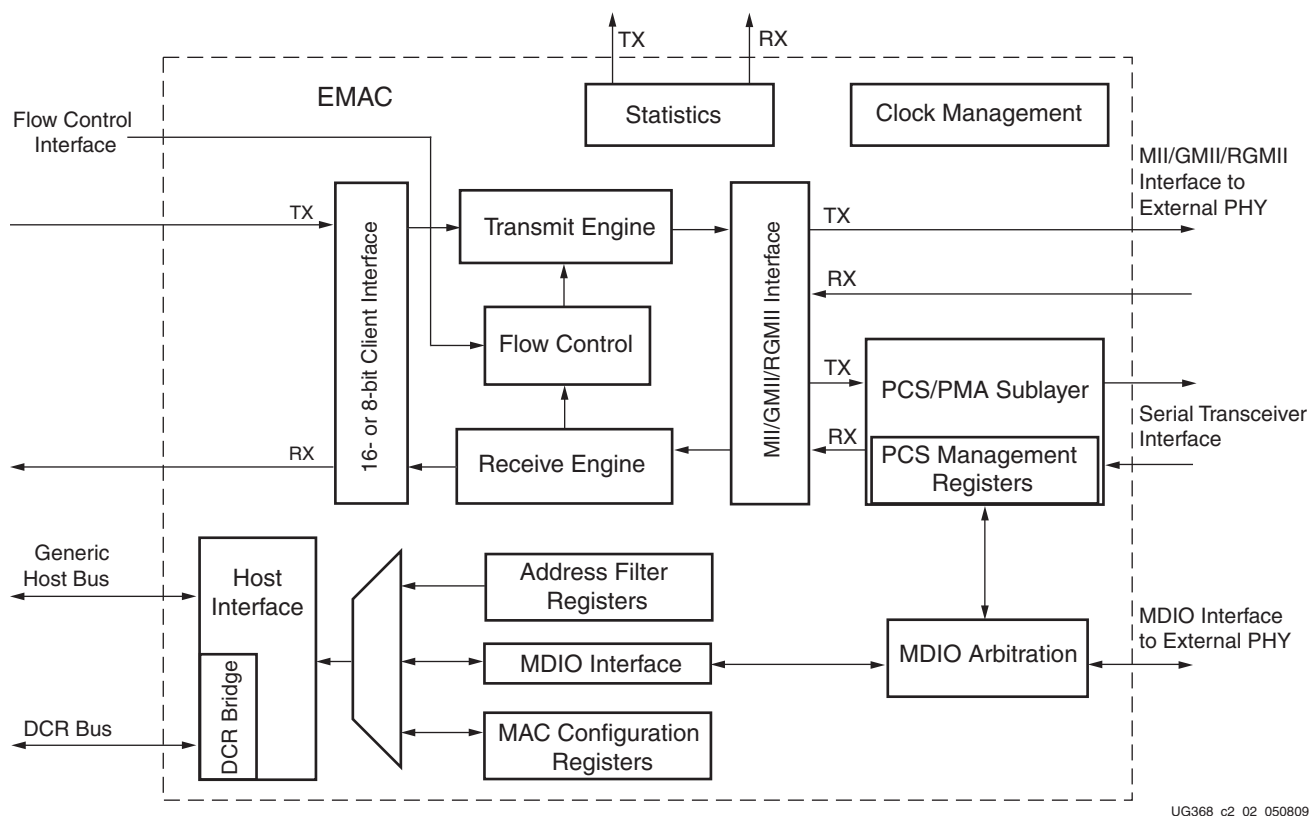
The Ethernet MAC has an optional Management Data I/O interface (MDIO), allowing access to the management registers of an external PHY and access to the physical interface



management registers within the Ethernet MAC (used only when it is configured in either 1000BASE-X or SGMII modes).

The Ethernet MAC outputs statistics vectors containing information about the Ethernet frames seen on its transmit and receive datapaths. The statistics vectors are multiplexed to reduce the number of pins at the block boundary. An external module (Statistics IP) can be implemented in the FPGA logic to accumulate all the TX and RX datapath statistics of the Ethernet MAC. Additional information on interfacing to the statistics block is provided in [Chapter 7, Interfacing to a Statistics Block](#).

A detailed functional block diagram of the Ethernet MAC is shown in [Figure 2-2](#).



**Figure 2-2: Ethernet MAC Functional Block Diagram (with Host Interface)**

The client interface shows the user transmit and receive interfaces connected to the transmit and receive engines of the Ethernet MAC, respectively. For detailed information on the protocol for data transmission and reception on this interface, refer to [Client Interface](#), page 45.

The flow control interface allows the client logic to force the physical layer to stop sending frames until the client is capable of accepting more. A description of how to use this interface from the client logic can be found in [Flow Control Block](#), page 69.

The Ethernet MAC has a programmable address filter to accept or reject incoming frames on the receive datapath. The full functionality of the Ethernet MAC address filter is described in [Address Filtering](#), page 66.



The host interface can access the MAC Configuration registers and the Address Filter registers. See [Host/DCR Bus Interfaces, page 81](#) to learn how to interface to any host via the generic host bus, or how to interface to a processor through the DCR bus.

The host interface can also initiate transactions on the MDIO interface, when the MDIO interface is enabled. This allows the host interface to access the PCS Management Registers within the Ethernet MAC, when it is configured in 1000BASE-X or SGMII mode. It is also possible for the host interface to initiate transactions via the MDIO interface to an external PHY device. When the MDIO interface is enabled and the host interface is disabled, an external device can still access the PCS Management Registers of the Ethernet MAC when it is configured in 1000BASE-X or SGMII mode. Information on these configurations and on using the MDIO protocol to access PCS Management registers can be found in [MDIO Interface, page 111](#).

On the physical interface of the Ethernet MAC, a common MII/GMII/RGMII interface block is shown. When the physical interface of the Ethernet MAC is configured to MII/GMII or RGMII, these signals can be used via standard I/Os to connect to an external physical interface. When the physical interface is configured in 1000BASE-X or SGMII mode, the PCS/PMA sublayer block in the Ethernet MAC is enabled. The PCS/PMA sublayer block interfaces directly to the serial transceiver. See [Physical Interface, page 133](#) for information on all physical interfaces, including clocking schemes and interfacing with serial transceivers.

The clock management module automatically configures the output clocks to the correct frequency based on the internal speed of the Ethernet MAC (10 Mb/s, 100 Mb/s, or 1000 Mb/s) and the Ethernet MAC mode settings (GMII, MII, RGMII, SGMII, and 1000BASE-X).

## Ethernet MAC Configuration Options

The Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC addresses all common configuration requirements for embedded Ethernet connectivity. This section provides an overview of the possible options for each Ethernet MAC. A full list of the mode configuration options is in [Table 2-16](#), and further details on the configuration options and how to set these using the CORE Generator™ tool can be found in [Accessing the Ethernet MAC from the CORE Generator Tool, page 21](#).

The choice of physical interface for the Ethernet MAC affects some of the other configuration options. [Table 2-1](#) provides a breakdown of compatible parameters.

Table 2-1: Ethernet MAC Block Static Configurations

Physical Interface	Client Interface		Speed				Clock Enable
	8-Bit	16-Bit	Tri-Speed	1G	10/100M	Overclocking up to 2.5G	
MII only	Y	N	N	N	Y	N	Y
GMII only	Y	N	N	Y	N	N	N
GMII/MII	Y	N	Y	Y	Y	N	Y (Tri-speed)
RGMII	Y	N	Y	Y	Y	N	Y (Tri-speed)
SGMII	Y	N	Y	Y	Y	N	N
1000BASE-X	Y	Y	N	Y	N	Y	N

A 16-bit client interface can only be used when the Ethernet MAC physical interface is configured to be 1000BASE-X. The 16-bit client interface allows the Ethernet MAC to run at an internal clock frequency of greater than 125 MHz, in support of operation up to 2.5 Gb/s data rate. This rate should not be used for IEEE Std 802.3-2005-compliant designs with but can be useful for backplane applications.

The possible speed configurations for the Ethernet MAC vary with the physical interface configuration. An initial value for the speed can be chosen for the Ethernet MAC at instantiation, and depending on the mode, this speed can later be modified by writing to the Ethernet MAC mode configuration register.

To minimize resource utilization, the Clock Enable advanced clocking scheme is available for specific physical interface configurations.

In addition to the configuration options listed in [Table 2-1](#), it is also possible to statically configure other interfaces:

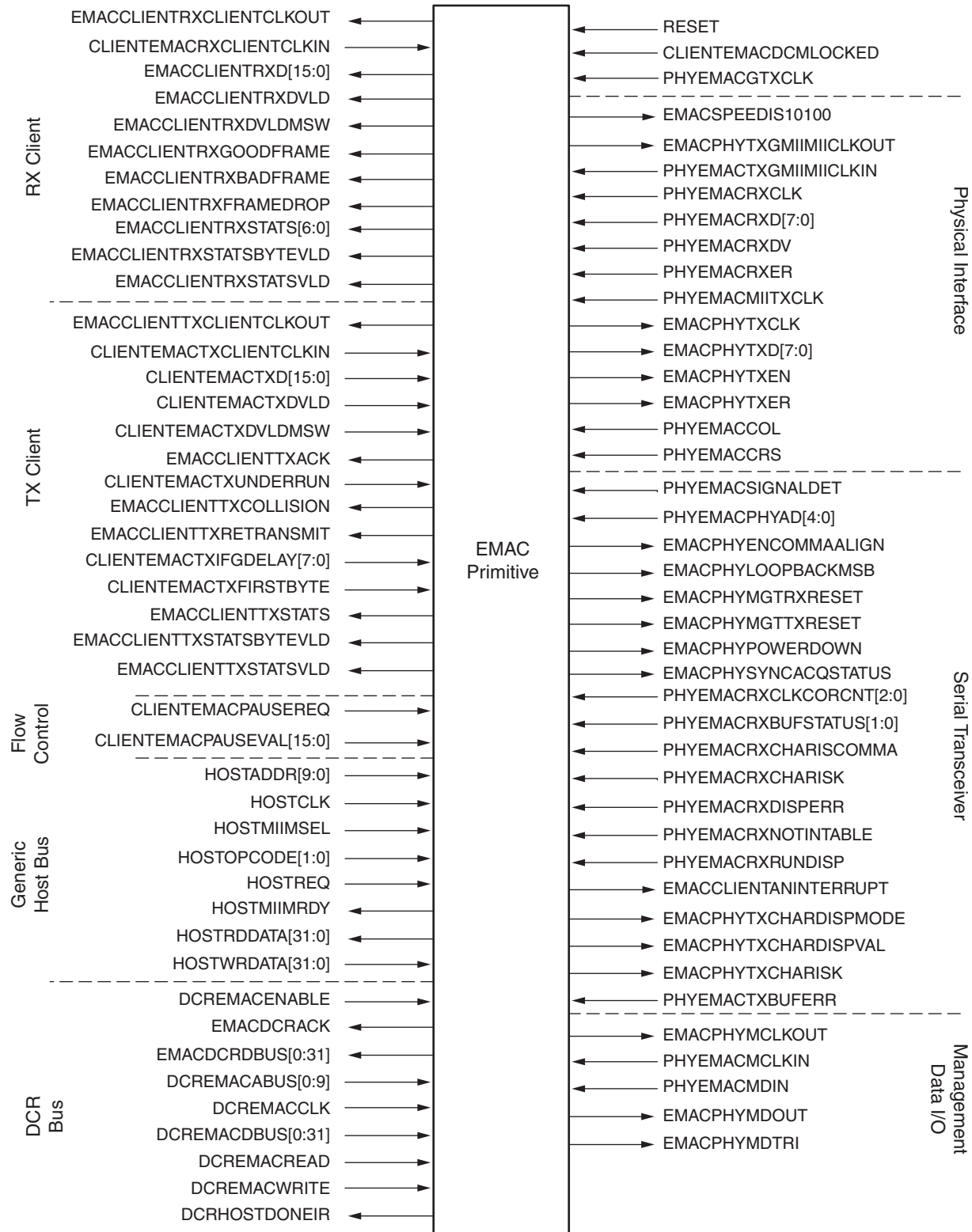
- The host interface can be enabled/disabled.
- The MDIO interface can be enabled/disabled.

## Ethernet MAC Primitive

The Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC is available to the Xilinx® software tools as a library primitive, named TEMAC\_SINGLE. This primitive encompasses the Ethernet MAC block as illustrated in [Figure 2-1](#).

The Ethernet MAC primitive can be simplified for specific customer needs by using the CORE Generator tool to create Ethernet MAC wrappers. [Accessing the Ethernet MAC from the CORE Generator Tool, page 21](#) describes how to use the CORE Generator tool to simplify the Ethernet MAC primitive.

[Figure 2-3](#) illustrates the Tri-Mode Ethernet MAC primitive.



UG368\_c2\_03\_050809

Figure 2-3: Ethernet MAC Primitive

## Ethernet MAC Signal Descriptions

This section defines all of the Ethernet MAC primitive signals. The signals are divided into the following categories:

- [Client-Side Signals](#)
- [Host Interface Signals](#)
- [Management Data Input/Output \(MDIO\) Signals](#)
- [MII/GMII/RGMII Physical Interface Signals](#)
- [Serial Transceiver Signals](#)
- [Global Clock and Reset Signals](#)

All the signals available in the Ethernet MAC primitive are described in this section.

### Client-Side Signals

#### TX Client Signals

[Table 2-2](#) describes the client-side transmit signals in the Ethernet MAC. These signals are used to transmit data from the client to the Ethernet MAC.

**Table 2-2: Transmit Client Interface Signals**

Signal	Direction	Description
CLIENTEMACTXD[15:0]	Input	Frame data for transmit. The datapath can be configured to be either 8 or 16 bits wide. Bits [7:0] are used for 8-bit width. The 16-bit interface is available only in 1000BASE-X PCS/PMA mode. See <a href="#">Transmit (TX) Client: 16-Bit Interface, page 54</a> .
CLIENTEMACTXDVLD	Input	Asserted by the client to indicate a valid data input at CLIENTEMACTXD[7:0].
CLIENTEMACTXDVLDMSW	Input	When the width of CLIENTEMACTXD is set to 16 bits, this signal denotes an odd number of bytes in the transmit datapath. In the frame with an odd number of bytes, the CLIENTEMACTXD[7:0] is valid on the last byte. When the width of CLIENTEMACTXD is set to 8 bits, this signal is connected to ground.
CLIENTEMACTXFIRSTBYTE	Input	This signal should be tied Low.
CLIENTEMACTXIFGDELAY[7:0]	Input	Configurable interframe gap (IFG) adjustment for full-duplex mode.
CLIENTEMACTXUNDERRUN	Input	Asserted by the client to force the Ethernet MAC to corrupt the current frame.
CLIENTEMACTXCLIENTCLKIN	Input	See <a href="#">Ethernet MAC Clocks, page 193</a> .
EMACCLIENTTXACK	Output	Handshake signal – Asserted when the Ethernet MAC accepts the first byte of data. On the next and subsequent rising clock edges, the client must provide the remainder of the frame data. See <a href="#">Normal Frame Transmission with 8-Bit Client Interface, page 47</a> .
EMACCLIENTTXCOLLISION	Output	Asserted by the Ethernet MAC to signal a collision on the medium. Any transmission in progress must be aborted. This signal is always deasserted in full-duplex mode.

Table 2-2: Transmit Client Interface Signals (Cont'd)

Signal	Direction	Description
EMACCLIENTTXRETRANSMIT	Output	Asserted by the Ethernet MAC at the same time as the EMACCLIENTTXCOLLISION signal. The client should resupply the aborted frame to the Ethernet MAC for retransmission. This signal is always deasserted in full-duplex mode.
EMACCLIENTTXSTATS	Output	The statistics data on the last data frame sent. The 32-bit TX raw statistics vector is output by one bit per cycle for statistics gathering. See <a href="#">Transmitter Statistics Vector</a> , page 74.
EMACCLIENTTXSTATSBYTEVLD	Output	Asserted if an Ethernet MAC frame byte is transmitted (including destination address to FCS). This is valid on every TX clock cycle.
EMACCLIENTTXSTATSVLD	Output	Asserted by the Ethernet MAC after a frame transmission to indicate a valid EMACCLIENTTXSTATS output. See <a href="#">Transmitter Statistics Vector</a> , page 74.
EMACCLIENTTXCLIENTCLKOUT	Output	See <a href="#">Ethernet MAC Clocks</a> , page 193.

## RX Client Signals

[Table 2-3](#) describes the client-side receive signals. These signals are used by the Ethernet MAC to transfer data to the client.

Table 2-3: Receive Client Interface Signals

Signal	Direction	Description
CLIENTEMACRXCLIENTCLKIN	Input	See <a href="#">Ethernet MAC Clocks</a> , page 193.
EMACCLIENTRXD[15:0]	Output	Frame data received from the Ethernet MAC. The datapath can be configured to either 8 bits or 16 bits wide. Bits [7:0] are used for 8-bit widths. The 16-bit interface is available only in 1000BASE-X PCS/PMA mode. See <a href="#">Receive (RX) Client: 16-Bit Interface</a> , page 64.
EMACCLIENTRXDVLD	Output	The Ethernet MAC indicates to the client the receipt of valid frame data.
EMACCLIENTRXFRAMEDROP	Output	This signal is asserted to notify the client that an incoming receive frames destination address does not match any addresses in the address filter. The signal functions even when the address filter is not enabled.
EMACCLIENTRXDVLDMSW	Output	This signal denotes an odd number of bytes in the receive datapath when the width of EMACCLIENTRXD is set to 16 bits. In a frame with an odd number of bytes, the EMACCLIENTRXD[7:0] byte is valid on the last byte. When the width of EMACCLIENTRXD is set to 8 bits, this signal should be left unconnected.
EMACCLIENTRXGOODFRAME	Output	This signal is asserted after the last receipt of data to indicate the reception of a compliant frame.
EMACCLIENTRXBADFRAME	Output	This signal is asserted after the last receipt of data to indicate the reception of a non-compliant frame.

Table 2-3: Receive Client Interface Signals (Cont'd)

Signal	Direction	Description
EMACCLIENTRXSTATS[6:0]	Output	The statistics data on the last received data frame. The 28-bit raw RX statistics vector is multiplexed into a 7-bits-per-RX clock cycle output for statistics gathering. See <a href="#">Receiver Statistics Vector</a> , page 77.
EMACCLIENTRXSTATSBYTEVLD	Output	Asserted if an Ethernet MAC frame byte (including destination address to FCS) is received. Valid on every RX clock cycle.
EMACCLIENTRXSTATSVLD	Output	Asserted by the Ethernet MAC after the end of receiving a frame to indicate a valid EMACCLIENTRXSTATS[6:0] output.
EMACCLIENTRXCLIENTCLKOUT	Output	See <a href="#">Ethernet MAC Clocks</a> , page 193.

### Flow Control Signals

[Table 2-4](#) describes the signals used by the client to request a flow control action from the transmit engine. The flow control block is designed per the specifications in IEEE Std 802.3-2005, clause 31. Flow control frames received by the Ethernet MAC are automatically handled.

Table 2-4: Flow Control Interface Signals

Signal	Direction	Description
CLIENTEMACPAUSEREQ	Input	Asserted by client to transmit a PAUSE frame.
CLIENTEMACPAUSEVAL[15:0]	Input	The amount of pause time for the transmitter as defined in the IEEE Std 802.3-2005 specification.

## Host Interface Signals

### Generic Host Bus Signals

[Table 2-5](#) outlines the host bus interface signals.

**Table 2-5: Generic Host Bus Signals**

Signal	Direction	Description
HOSTCLK	Input	Clock supplied for running the host.
HOSTOPCODE[1:0]	Input	Defines the operation to be performed over the MDIO interface. Bit 1 is also used in configuration register access. See <a href="#">Using the Host Bus, page 92</a> .
HOSTADDR[9:0]	Input	Address of register to be accessed.
HOSTWRDATA[31:0]	Input	Data bus to write to register.
HOSTRDDATA[31:0]	Output	Data bus to read from register.
HOSTMIIMSEL	Input	When asserted, the MDIO interface is accessed. When deasserted, the Ethernet MAC internal configuration registers are accessed.
HOSTREQ	Input	Used to signal a transaction on the MDIO interface.
HOSTMIIMRDY	Output	When High, the MDIO interface has completed any pending transaction and is ready for a new transaction.

**Notes:**

1. All signals are synchronous to HOSTCLK and are active High.
2. When using a host processor with the DCR bus, the host bus signals are used to read the optional FPGA logic-based statistics registers. See [Chapter 7, Interfacing to a Statistics Block](#).

## DCR Bus Signals

Table 2-6 outlines the DCR bus interface signals.

Table 2-6: DCR Bus Signals

Signal	Direction	Description
DCREMACCLK	Input	Clock for the DCR interface.
DCREMACABUS[0:9]	Input	DCR address bus. Bits 0 through 7 must be driven to match DCR base address of one of the Ethernet MACs, as set by the EMAC_DCRBASEADDR[7:0] attribute. See Table 2-20, page 43 for more information on this attribute.
DCREMACREAD	Input	DCR read request.
DCREMACWRITE	Input	DCR write request.
DCREMACDBUS[0:31]	Input	DCR write data bus.
DCREMACENABLE	Input	When using the DCR bus, this signal is asserted High. When using the host bus interface, the signal is asserted Low.
EMACDCRDBUS[0:31]	Output	DCR read data bus.
EMACDCRACK	Output	DCR acknowledge.
DCRHOSTDONEIR	Output	Interrupt signal to indicate when the Ethernet MAC register access is complete.

## Management Data Input/Output (MDIO) Signals

Table 2-7 describes the Management Data Input/Output (MDIO) interface signals. The MDIO format is defined in IEEE Std 802.3-2005, clause 22.

Table 2-7: MDIO Interface Signals

Signal	Direction	Description
EMACPHYMCLKOUT	Output	Management clock derived from the host clock or PHYEMACMCLKIN.
PHYEMACMCLKIN	Input	When the host is not used, access to the PCS must be provided by an external MDIO controller. In this situation, the management clock is an input to the EMAC block.
PHYEMACMDIN	Input	Signal from the physical interface for communicating the configuration and status. If unused, must be tied High.
EMACPHYMDOUT	Output	Signal to output the configuration and command to the physical interface.
EMACPHYMDTRI	Output	The three-state control to accompany EMACPHYMDOUT.

## MII/GMII/RGMII Physical Interface Signals

The Ethernet MAC has several signals that change definition depending on the selected physical interface configuration. This section describes the physical interface signals when the Ethernet MAC physical interface configuration is set to either MII, GMII, or RGMII.



## Data and Control Signals

Table 2-8 shows the data and control signals for the different physical interface modes. The functionality of each of these signals is defined after the static configuration mode is set. This is achieved via attributes, as described in [Ethernet MAC Attributes](#), page 36.

Table 2-8: PHY Data and Control Signals

Signal	Direction	Mode	Description
EMACPHYTXEN	Output	GMII	The data enable control signal to the PHY.
		RGMII	
		RGMII	The RGMII_TX_CTL_RISING signal to the PHY.
EMACPHYTXER	Output	10/100 MII	The error control signal to the PHY.
		GMII	
		RGMII	The RGMII_TX_CTL_FALLING signal to the PHY.
EMACPHYTXD[7:0]	Output	10/100 MII	EMACPHYTXD[3:0] is the transmit data bus to the PHY. EMACPHYTXD[7:4] are driven Low.
		GMII	The transmit data signal to the PHY.
		RGMII	EMACPHYTXD[3:0] are the RGMII_TXD_RISING signals and EMACPHYTXD[7:4] are the RGMII_TXD_FALLING signals to the PHY.
		SGMII	The TX_DATA signal to the serial transceiver.
		1000BASE-X	
PHYEMACRXDV	Input	10/100 MII	The data valid control signal from the PHY.
		GMII	
		RGMII	The RGMII_RX_CTL_RISING signal.
		SGMII	The RXREALIGN signal from the serial transceiver.
		1000BASE-X	
PHYEMACRXER	Input	10/100 MII	The error control signal from the PHY.
		GMII	
		RGMII	The RGMII_RX_CTL_FALLING signal from the PHY.
PHYEMACRXD[7:0]	Input	10/100 MII	PHYEMACRXD[3:0] are the received data signals from the PHY. PHYEMACRXD[7:4] are left unconnected.
		GMII	The received data signal to the PHY.
		RGMII	PHYEMACRXD[3:0] are the RGMII_RXD_RISING signals and PHYEMACRXD[7:4] are the RGMII_RXD_FALLING signals from the PHY.
		SGMII	The RX_DATA signal from the serial transceiver.
		1000BASE-X	

Table 2-8: PHY Data and Control Signals (Cont'd)

Signal	Direction	Mode	Description
PHYEMACCOL	Input	10/100 MII	The collision control signal from the PHY, used in half-duplex mode.
		SGMII	Unused, and tied Low.
		1000BASE-X	
PHYEMACCRS	Input	10/100 MII	The carrier sense control signal from the PHY, used in half-duplex mode.

The usage of the Ethernet MAC clock signals associated with the physical interface also changes, based on the selected configuration. This is fully described in [Appendix B, Ethernet MAC Clocks](#). Table 2-9 lists the physical interface clock signals and references the associated descriptions of each clock.

Table 2-9: PHY Clock Signals

Signal	Direction	Description
PHYEMACMIITXCLK	Input	See <a href="#">PHYEMACMIITXCLK</a> , page 199.
EMACPHYTXCLK	Output	See <a href="#">EMACPHYTXCLK</a> , page 201.
EMACPHYTXGMIIMIICLKOUT	Output	See <a href="#">EMACPHYTXGMIIMIICLKOUT</a> , <a href="#">PHYEMACTXGMIIMIICLKIN</a> , page 200.
PHYEMACTXGMIIMIICLKIN	Input	See <a href="#">EMACPHYTXGMIIMIICLKOUT</a> , <a href="#">PHYEMACTXGMIIMIICLKIN</a> , page 200.
PHYEMACRXCLK	Input	See <a href="#">PHYEMACRXCLK</a> , page 200.

Table 2-10 shows Ethernet MAC output that indicates the current operating speed of the Ethernet MAC.

Table 2-10: Physical Interface Speed Signal

Signal	Direction	Description
EMACSPEDIS10100	Output	When asserted High, the physical interface is operating at 10 Mb/s or 100 Mb/s. When asserted Low, the physical interface is operating at 1 Gb/s.

## Serial Transceiver Signals

Table 2-11 shows the signals used to connect the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC to the serial transceiver (see the *Virtex-6 FPGA GTX Transceiver User Guide*).

Table 2-11: Serial Transceiver Connections

Signal	Direction	Description
EMACPHYENCOMMAALIGN	Output	Enable serial transceiver PMA layer to realign to commas.
EMACPHYLOOPBACKMSB	Output	Perform loopback testing in the serial transceiver from TX to RX.
EMACPHYMGTRXRESET	Output	Reset to serial transceiver RXRESET.
EMACPHYMGTTXRESET	Output	Reset to serial transceiver TXRESET.

Table 2-11: Serial Transceiver Connections (Cont'd)

Signal	Direction	Description
EMACPHYPOWERDOWN	Output	Power-down the serial transceiver.
EMACPHYSYNACQSTATUS	Output	The output from the receiver's synchronization state machine of IEEE Std 802.3-2005, clause 36. When asserted High, the received bitstream is synchronized. The state machine is in one of the SYNC_ACQUIRED states of IEEE Std 802.3-2005, figures 36-39. When deasserted Low, no synchronization has been obtained.
EMACPHYTXCHARDISPMODE	Output	Set running disparity for current byte.
EMACPHYTXCHARDISPVAL	Output	Set running disparity value.
EMACPHYTXCHARISK	Output	K character transmitted in TXDATA.
PHYEMACRXBUFSTATUS[1:0]	Input	Receiver Elastic Buffer Status: Bit 1 asserted indicates overflow or underflow.
PHYEMACRXCHARISCOMMA	Input	Comma detected in RXDATA.
PHYEMACRXCHARISK	Input	K character received or extra data bit in RXDATA. When RXNOTINTABLE is asserted, this signal becomes the tenth bit in RXDATA.
PHYEMACRXDISPERR	Input	Disparity error in RXDATA.
PHYEMACRXNOTINTABLE	Input	Indicates non-existent 8B/10 code.
PHYEMACRXRUNDISP	Input	Running disparity in the received serial data. When RXNOTINTABLE is asserted in RXDATA, this signal becomes the ninth data bit.
PHYEMACRXCLKCORCNT[2:0]	Input	Status showing the occurrence of a clock correction.
PHYEMACTXBUFERR	Input	TX buffer error (overflow or underflow).

Table 2-12 shows the PCS/PMA signals.

Table 2-12: PCS/PMA Signals

Signal	Direction	Description
PHYEMACPHYAD[4:0]	Input	Physical interface address of MDIO register set for the PCS sublayer.
PHYEMACSIGNALEDIT	Input	Signal direct from PMD sublayer indicating the presence of light detected at the optical receiver, as defined in IEEE Std 802.3-2005, clause 36. If asserted High, the optical receiver has detected light. If deasserted Low, indicates the absence of light. If unused, this signal should be tied High for correct operation.
EMACCLIENTANINTERRUPT	Output	Interrupt upon auto-negotiation.

## Global Clock and Reset Signals

### Global Clock Signal

[Table 2-13](#) shows the global clock signal that is necessary for most configurations of the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC. See [Appendix B, Ethernet MAC Clocks](#) for further details.

**Table 2-13: Global Clock Signals**

Signal	Direction	Description
PHYEMACGTCLK	Input	Clock supplied by the user to derive the other transmit clocks. Clock tolerance must be within the IEEE Std 802.3-2005 specification.

### Reset Signal

[Table 2-14](#) describes the Reset signal.

**Table 2-14: Reset Signal**

Signal	Direction	Description
Reset	Input	Asynchronous reset of the entire Ethernet MAC and associated logic in the wrapper files.

### CLIENTEMACDCMLOCKED Signal

[Table 2-15](#) describes the CLIENTEMACDCMLOCKED signal.

**Table 2-15: CLIENTEMACDCMLOCKED Signal**

Signal	Direction	Description
CLIENTEMACDCMLOCKED	Input	If a MMCM is used to derive any of the clock signals, the LOCKED port of the MMCM must be connected to the CLIENTEMACDCMLOCKED port. The Ethernet MAC is held in reset until CLIENTEMACDCMLOCKED is asserted High. If the serial transceiver PLL is used, the PLLKDET signal should be ANDed with the MMCM LOCK signal. If an MMCM is not used, CLIENTEMACDCMLOCKED must be tied High.

## Ethernet MAC Attributes

This section describes the attributes (control parameters) that are used to configure the Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC. Some of these attributes control the static configuration of the Ethernet MAC and others are used to preconfigure the internal control registers. The values of these attributes are loaded into the Ethernet MAC at power-up or when the Ethernet MAC is reset. An attribute value of TRUE is translated into a bit value of 1; a value of FALSE equals a bit value of 0.

The attributes are divided into five sections: mode configuration, MAC configuration, physical interface, address filter, and DCR base address attributes.

The mode configuration attributes control the static behavior of the Ethernet MAC, including whether the Ethernet MAC has a host interface, the type of physical interface,

the width of the client interface and whether the MDIO interface is enabled. These attributes generally do not support dynamic modification.

All other attributes described in [Table 2-16](#) (unless stated otherwise) define default values for internal registers. If the host interface is not enabled on the Ethernet MAC, these attributes allow the user to set the internal control register defaults. If the host interface is enabled (by setting the EMAC\_HOST\_ENABLE attribute to be TRUE), then the host interface can be used to dynamically change the associated register contents or to read the register contents.

**Table 2-16: Mode Configuration Attributes**

Attribute	Type	Description
Attribute to Configure Default Register Values for Management Configuration Register		See <a href="#">Table 4-8, page 88</a> for register details. The MDIO interface can be enabled/disabled using the host interface, when the host interface is enabled. (The host interface is enabled by setting the EMAC_HOST_ENABLE attribute to TRUE). If the host interface is not enabled then the MDIO enable value is fixed by this attribute.
EMAC_MDIO_ENABLE	Boolean	When EMAC_HOST_ENABLE is FALSE, setting this attribute to TRUE enables the use of MDIO in the Ethernet MAC. When EMAC_HOST_ENABLE and EMAC_MDIO_ENABLE are both TRUE, the MDIO interface is enabled when the MDIO enable bit is set in the Ethernet MAC Management Configuration Register. See <a href="#">MDIO Interface, page 111</a> .
Attributes to Configure Fixed Mode for the Ethernet MAC		See <a href="#">Table 4-6, page 86</a> for register details.
EMAC_HOST_ENABLE	Boolean	Setting this attribute to TRUE enables the use of the Ethernet MAC host interface via either the generic host bus or the DCR bus.
EMAC_TX16BITCLIENT_ENABLE	Boolean	When this attribute is set to TRUE, the TX client data interface is 16 bits wide. When this attribute is set to FALSE, the TX client data interface is 8 bits wide.
EMAC_RX16BITCLIENT_ENABLE	Boolean	When this attribute is set to TRUE, the RX client data interface is 16 bits wide. When this attribute is set to FALSE, the RX client data interface is 8 bits wide.
Attributes to Configure Fixed Physical Interface for the Ethernet MAC		The following three attributes define the physical interface of the Ethernet MAC. These attributes are mutually exclusive. If none of these three attributes is set to TRUE, then the physical interface is enabled for 10/100 MII and GMII modes. See <a href="#">Table 4-6, page 86</a> for register details. When EMAC_SGMII_ENABLE is set to TRUE, run-time switching between SGMII and 1000BASE-X physical interface modes is supported. See <a href="#">SGMII and 1000BASE-X PCS/PMA (8-Bit Data Client) Modes, page 200</a> .
EMAC_RGMII_ENABLE	Boolean	Setting this attribute to TRUE enables the Ethernet MAC for RGMII mode.
EMAC_SGMII_ENABLE	Boolean	Setting this attribute to TRUE enables the Ethernet MAC for SGMII mode.
EMAC_1000BASEX_ENABLE	Boolean	Setting this attribute to TRUE enables the Ethernet MAC for 1000BASE-X mode.

Table 2-16: Mode Configuration Attributes (Cont'd)

Attribute	Type	Description
Attributes to Configure Speed for the Ethernet MAC		<p>These attributes determine the speed of the Ethernet MAC after reset or power-up.</p> <p>The speed of the Ethernet MAC can be changed in multi-speed configurations using the host interface, when the host interface is enabled. (The host interface is enabled by setting the EMAC_HOST_ENABLE attribute to TRUE). If the host interface is not enabled, then the speed of the Ethernet MAC is directly set by these two attributes.</p> <p>(EMAC_SPEED_MSB: EMAC_SPEED_LSB)</p> <p>1 : 0 = 1000 Mb/s</p> <p>0 : 1 = 100 Mb/s</p> <p>0 : 0 = 10 Mb/s</p> <p>1 : 1 = not applicable</p> <p>See <a href="#">Ethernet MAC Register Addresses, page 83</a> for register details.</p>
EMAC_SPEED_MSB	Boolean	EMAC_SPEED[1]. (TRUE takes the bit value 1 in the table cell above.)
EMAC_SPEED_LSB	Boolean	EMAC_SPEED[0]. (TRUE takes the bit value 1 in the table cell above.)
Attributes to Configure Advanced Clocking Scheme for the Ethernet MAC		To reduce the number of BUFs used for the Ethernet MAC, an optimized clocking mode is available. See <a href="#">Clock Connections to and from FPGA Logic, page 195</a> for details.
EMAC_USECLKEN	Boolean	When this attribute is set to TRUE, the client interface clock output is switched to clock enable, thus eliminating BUFs on the client interface. See <a href="#">The Clock Enable Advanced Clocking Scheme, page 197</a> for an introduction to this clocking mode option.

Table 2-17: MAC Configuration Attributes

Attribute	Type	Description
Attribute to Configure Default Register Value for the Address Filter Mode Register		<p>The address filter can be enabled/disabled using the host interface, when the host interface is enabled. (The host interface is enabled by setting the EMAC_HOST_ENABLE attribute to TRUE). If the host interface is not enabled, the address filter mode is fixed by the following attribute value.</p> <p>See <a href="#">Table 4-13, page 92</a> for register details.</p>
EMAC_ADDRFILTER_ENABLE	Boolean	<p>Address Filter Enable: This attribute sets the default value of the Promiscuous Mode enable bit in the Address Filter Mode register.</p> <p>Setting this attribute to TRUE enables the use of the address filter module in the Ethernet MAC.</p>
Attributes to Configure Default Register Value for the Flow Control Configuration Register		The following two attributes configure the Ethernet MAC flow control module. See <a href="#">Table 4-5, page 86</a> for register details.

Table 2-17: MAC Configuration Attributes (Cont'd)

Attribute	Type	Description
EMAC_RXFLOWCTRL_ENABLE	Boolean	<p>Receive Flow Control Enable: This attribute sets the default value of the Flow control enable (RX) bit in the Flow Control Configuration Register.</p> <p>When this attribute is set to TRUE and full-duplex mode is enabled, receive flow control is enabled. Any flow control frames received by the Ethernet MAC cause the transmitter operation to be inhibited. When this attribute is set to FALSE, any flow control frames received by the Ethernet MAC are passed on to the client. See <a href="#">Flow Control Block, page 69</a> for description of flow control functionality.</p>
EMAC_TXFLOWCTRL_ENABLE	Boolean	<p>Transmit Flow Control Enable: This attribute sets the default value of the Flow control enable (TX) bit in the Flow Control Configuration Register.</p> <p>When this attribute is set to TRUE and full-duplex mode is enabled, asserting the CLIENTEMACPAUSEREQ signal causes the Ethernet MAC to send a flow control frame out from the transmitter.</p> <p>When this attribute is set to FALSE, asserting the CLIENTEMACPAUSEREQ signal has no effect.</p>
Attributes to Configure Default Register Value for the Transmitter Configuration Register		The following attributes configure the transmit engine of the Ethernet MAC. See <a href="#">Transmitter Configuration Register, page 85</a> for register details.
EMAC_TXRESET	Boolean	<p>This attribute sets the default value of the Reset bit in the Transmitter Configuration Register.</p> <p>When this attribute is set to TRUE, the Ethernet MAC transmitter is reset. This reset also sets all of the transmitter configuration registers to their default values.</p>
EMAC_TXJUMBOFRAME_ENABLE	Boolean	<p>This attribute sets the default value of the Jumbo Frame Enable bit in the Transmitter Configuration Register.</p> <p>When this attribute is set to TRUE, the Ethernet MAC transmitter allows frames larger than the maximum legal frame length specified in IEEE Std 802.3-2005 to be sent. When FALSE, the Ethernet MAC transmitter only allows frames up to the legal maximum to be sent.</p>
EMAC_TXINBANDFCS_ENABLE	Boolean	<p>This attribute sets the default value of the In-Band FCS Enable bit in the Transmitter Configuration Register.</p> <p>When this attribute is set to TRUE, the Ethernet MAC transmitter expects the FCS field to be passed in by the client. When FALSE, the Ethernet MAC transmitter appends padding as required, computes the FCS, and appends it to the frame.</p>
EMAC_TX_ENABLE	Boolean	<p>This attribute sets the default value of the Transmit Enable bit in the Transmitter Configuration Register.</p> <p>When this attribute is set to TRUE, the transmitter is operational. When FALSE, the transmitter is disabled.</p>



Table 2-17: MAC Configuration Attributes (Cont'd)

Attribute	Type	Description
EMAC_TXVLAN_ENABLE	Boolean	This attribute sets the default value of the VLAN Enable bit in the Transmitter Configuration Register. When this attribute is set to TRUE, the transmitter allows the transmission of VLAN tagged frames.
EMAC_TXHALFDUPLEX	Boolean	This attribute sets the default value of the Half Duplex mode bit in the Transmitter Configuration Register. When this attribute is set to TRUE, the transmitter operates in half-duplex mode. When FALSE, the transmitter operates in full-duplex mode.
EMAC_TXIFGADJUST_ENABLE	Boolean	This attribute sets the default value of the IFG Adjustment Enable bit in the Transmitter Configuration Register. When this attribute is set to TRUE and full-duplex mode is enabled, the transmitter reads the value of the CLIENTEMACTXIFGDELAY[7:0] port and sets the IFG accordingly. When FALSE, the transmitter always inserts at least the legal minimum IFG.
Attributes to Configure Default Register Value for the Receiver Configuration Registers		Configures the receive engine of the Ethernet MAC. See <a href="#">Table 4-2, page 84</a> and <a href="#">Table 4-3, page 84</a> for register details.
EMAC_RXRESET	Boolean	This attribute sets the default value of the Reset bit in the Receiver Configuration Register. When this attribute is set to TRUE, the receiver is held in reset. This signal is an input to the reset circuit for the receiver block.
EMAC_RXJUMBOFRAME_ENABLE	Boolean	This attribute sets the default value of the Jumbo Frame Enable bit in the Receiver Configuration Register. When this attribute is set to FALSE, the receiver does not pass frames longer than the maximum legal frame size specified in IEEE Std 802.3-2005. When TRUE, the receiver does not have an upper limit on frame size.
EMAC_RXINBANDFCS_ENABLE	Boolean	This attribute sets the default value of the In-band FCS Enable bit in the Receiver Configuration Register. When this attribute is set to TRUE, the Ethernet MAC receiver passes the FCS field up to the client. When FALSE, the Ethernet MAC receiver does not pass the FCS field. In both cases, the FCS field is verified on the frame.
EMAC_RX_ENABLE	Boolean	This attribute sets the default value of the Receive Enable bit in the Receiver Configuration Register. When this attribute is set to TRUE, the receiver block is operational. When FALSE, the block ignores activity on the physical interface RX port.
EMAC_RXVLAN_ENABLE	Boolean	This attribute sets the default value of the VLAN Enable bit in the Receiver Configuration Register. When this attribute is set to TRUE, VLAN tagged frames are accepted by the receiver.



Table 2-17: MAC Configuration Attributes (Cont'd)

Attribute	Type	Description
EMAC_RXHALFDUPLEX	Boolean	This attribute sets the default value of the Half Duplex mode bit in the Receiver Configuration Register. When this attribute is set to TRUE, the receiver operates in half-duplex mode. When FALSE, the receiver operates in full-duplex mode.
EMAC_LTCHECK_DISABLE	Boolean	This attribute sets the default value of the Length/Type Check Disable bit in the Receiver Configuration register. Setting this attribute to TRUE disables the Length/Type check on received frames where the Length/Type field is less than 0x0600. See <a href="#">Length/Type Field Error Checks, page 61</a> for more information.
EMAC_CTRLLENCHECK_DISABLE	Boolean	This attribute sets the default value for the Control Frame Length Check Disable bit in the Receiver Configuration Register. Setting this attribute to TRUE allows control frames, designated by a Length/Type field value of 0x8808, which are larger than 64 bytes in length, to be accepted by the receiver block.
EMAC_PAUSEADDR[47:0]	48-bit Binary	This attribute sets the default value of the Pause Frame Ethernet MAC Address [47:0] in the Receiver Configuration (Word 0) and (Word 1) registers. This address is used by the Ethernet MAC to match against the destination address of any incoming flow control frames and as the source address for any outbound flow control frames. The address is ordered for the least significant byte in the register to have the first byte transmitted or received; for example, an Ethernet MAC address of AA-BB-CC-DD-EE-FF is stored in byte [47:0] as 0xFFEEDDCCBBAA. The Pause frame address should be set to the same Ethernet MAC address as the EMAC_UNICASTADDR[47:0] attributes.

Table 2-18: Physical Interface Attributes

Attribute	Type	Description
Attributes to Configure Default Register Values for SGMII or 1000BASE-X Management Registers		The following attributes are only used in SGMII or 1000BASE-X modes. See <a href="#">1000BASE-X PCS/PMA Management Registers, page 118</a> and <a href="#">SGMII Management Registers, page 125</a> for register details. See <a href="#">MDIO Implementation in the Ethernet MAC, page 115</a> for information on how to access these registers.
EMAC_PHYRESET	Boolean	This attribute sets the default value of the Reset bit in the PCS/PMA Management Register 0 (Control Register). A value of TRUE resets the PCS/PMA module.
EMAC_PHYINITAUTONEG_ENABLE	Boolean	This attribute sets the default value of the auto-negotiation enable bit in the PCS/PMA Management Register 0 (Control Register). A value of TRUE enables auto-negotiation of the PCS/PMA module.

Table 2-18: Physical Interface Attributes (Cont'd)

Attribute	Type	Description
EMAC_PHYISOLATE	Boolean	This attribute sets the default value of the Isolate bit in the PCS/PMA Management Register 0 (Control Register). A value of TRUE causes the PCS/PMA sublayer logic to behave as if it is electrically isolated from the attached Ethernet MAC, as defined in IEEE Std 802.3-2005, clause 22.2.4.1.6. Therefore, frames transmitted by the Ethernet MAC are not forwarded through the PCS/PMA. Frames received by the PCS/PMA are not relayed to the Ethernet MAC.
EMAC_PHYPOWERDOWN	Boolean	This attribute sets the default value of the Power Down bit in the PCS/PMA Management Register 0 (Control Register). A value of TRUE causes the serial transceiver to be placed in a Low power state. A reset must be applied to clear the Low power state.
EMAC_PHYLOOPBACKMSB	Boolean	This attribute sets the default value of the Loopback bit in the PCS/PMA Management Register 0 (Control Register). A value of TRUE for this attribute enables loopback. EMAC_GTLOOPBACK sets if the loopback occurs in the Ethernet MAC or in the serial transceiver.
EMAC_UNIDIRECTION_ENABLE	Boolean	This attribute sets the default value of the unidirection enable bit in the PCS/PMA Management Register 0 (Control Register). Setting this attribute to TRUE allows the Ethernet MAC to transmit regardless of whether a valid link has been established.
EMAC_GTLOOPBACK	Boolean	<p>This attribute sets the default value of the GT Loopback bit in <a href="#">Vendor-Specific Register: Loopback Control Register (Register 17)</a>, <a href="#">page 124</a>. This attribute controls where the loopback happens when loopback is selected.</p> <p>When EMAC_GTLOOPBACK is TRUE, the loopback occurs in the serial transceiver. Near-End parallel PCS loopback is used.</p> <p>When EMAC_GTLOOPBACK is FALSE, the loopback is internal to the EMAC. In this state, idles are transmitted to the serial transceiver.</p>
EMAC_MDIO_IGNORE_PHYADZERO	Boolean	Setting this attribute to TRUE prevents the Ethernet MAC's internal PCS/PMA Management Registers from responding to MDIO transactions when PHYAD equals 0, defined by IEEE Std 802.3-2005 as the MDIO broadcast address. This attribute does not affect MDIO devices external to the Ethernet MAC.
Attribute to Configure Auto-Negotiation Parameters		<p>See the following sections for details on attribute value settings for Physical interface settings of SGMII or 1000BASE-X:</p> <ul style="list-style-type: none"> <li>• <a href="#">SGMII Auto-Negotiation Link Timer</a>, <a href="#">page 183</a>.</li> <li>• <a href="#">1000BASE-X Auto-Negotiation Link Timer</a>, <a href="#">page 166</a>.</li> </ul>
EMAC_LINKTIMERVAL[8:0]	9-bit Binary	<p>The default value for the programmable auto-negotiation link timer.</p> <p>This value can be changed through the host interface.</p>

Table 2-19: Address Filter Attributes

Attribute	Type	Description
Attribute to Configure Default Register Value for the Unicast Address Register		This attribute sets the default value of the Unicast Address[47:0] in the Unicast Address (Word 0) and (Word 1) registers. See <a href="#">Table 4-9, page 90</a> and <a href="#">Table 4-10, page 90</a> for register details.
EMAC_UNICASTADDR[47:0]	48-bit Binary	<p>This 48-bit attribute is used to set the Ethernet MAC unicast address used by the address filter block to see if the incoming frame is destined for the Ethernet MAC.</p> <p>The address is ordered for the least significant byte in the register to have the first byte transmitted or received; for example, an Ethernet MAC address of 06-05-04-03-02-01 is stored in bits [47:0] as 0x010203040506.</p> <p>This default value for the unicast address can be overridden by software through the host interface.</p>

Table 2-20: DCR Base Address Attributes

Attribute	Type	Description
Attribute to Configure Default Value for DCR Base Address		See <a href="#">Using the DCR Bus, page 96</a> for a description of how the DCR Base Addresses are used to access different registers.
EMAC_DCRBASEADDR[7:0]	8-bit Binary	This attribute allows separate DCR base addresses to be used for each Ethernet MAC.



## Client Interface

---

This chapter provides useful design information for the Virtex®-6 FPGA Embedded Tri-Mode Ethernet MAC. It contains the following sections:

- [Transmit \(TX\) Client: 8-Bit Interface \(without Clock Enables\)](#)
- [Transmit \(TX\) Client: 8-Bit Interface \(with Clock Enables\)](#)
- [Transmit \(TX\) Client: 16-Bit Interface](#)
- [Receive \(RX\) Client: 8-Bit Interface \(without Clock Enables\)](#)
- [Receive \(RX\) Client: 8-Bit Interface \(with Clock Enables\)](#)
- [Receive \(RX\) Client: 16-Bit Interface](#)
- [Address Filtering](#)
- [Flow Control Block](#)
- [Statistics Vectors](#)

The client interface is designed for maximum flexibility for matching the client switching logic or network processor interface.

The transmit and receive client interfaces can be configured to handle either 8-bit data transfers or 16-bit data transfers, where the default is 8 bits.

The 8-bit client operation supports all physical interfaces and is offered in two clocking configurations: the standard clocking scheme without clock enables or an alternative clock enable scheme. [Ethernet MAC Clocks, page 193](#) introduces these two clocking models and describes how the clock enable scheme can be used to reduce the number of global clock buffers required for the design. [Table 2-1, page 25](#) illustrates the physical interface configurations that can use the advanced clock enable scheme.

The 16-bit client operation is enabled by setting the EMAC\_RX16BITCLIENT\_ENABLE and EMAC\_TX16BITCLIENT\_ENABLE attributes to TRUE. This mode of operation is only available with the 1000BASE-X PCS/PMA physical interface. Using this scheme, the Ethernet MAC can be overclocked to provide operation above 1 Gb/s. [16-Bit Data Client, page 162](#) provides a description of the clocking scheme for this mode.

[Table 3-1](#) defines the abbreviations used throughout this chapter.

**Table 3-1: Abbreviations Used in this Chapter**

Abbreviation	Definition	Length
DA	Destination address	6 bytes
SA	Source address	6 bytes
L/T	Length/Type field	2 bytes
FCS	Frame check sequences	4 bytes

## Transmit (TX) Client: 8-Bit Interface (without Clock Enables)

This section covers the client transmit interface when it is configured to be 8 bits wide and when the default clocking scheme is used. [Ethernet MAC Attributes, page 36](#) describes the EMAC\_USECLKEN attribute, which is set to FALSE for this configuration. In this configuration, CLIENTEMACTXD[15:8] and CLIENTEMACTXDVLDMWS must be tied to ground because the upper 8 bits of the data bus are not used.

See [Ethernet MAC Clocks, page 193](#) for some general information on this standard clocking scheme. More details are available for specific physical interface configurations in the following sections:

- MII
  - [MII Standard Clock Management](#)
- GMII
  - [GMII Clock Management for 1 Gb/s Only](#)
  - [GMII Standard Clock Management for Tri-Speed Operation](#)
- RGMII
  - [RGMII Clock Management for 1 Gb/s Only](#)
  - [RGMII Standard Clock Management for Tri-Speed Operation](#)
- SGMII
  - [SGMII Clock Management](#)
- 1000BASE-X
  - [1000BASE-X PCS/PMA Clock Management](#)

## Normal Frame Transmission with 8-Bit Client Interface

The timing of a normal outbound frame transfer is shown in [Figure 3-1](#). When the client transmits a frame, the first column of data is placed on the CLIENTEMACTXD[7:0] port, and CLIENTEMACTXDVLD is asserted High. After the Ethernet MAC reads the first byte of data, it asserts the EMACCLIENTTXACK signal. On subsequent rising clock edges, the client must provide the rest of the frame data. CLIENTEMACTXDVLD is deasserted Low to signal an end-of-frame to the Ethernet MAC.

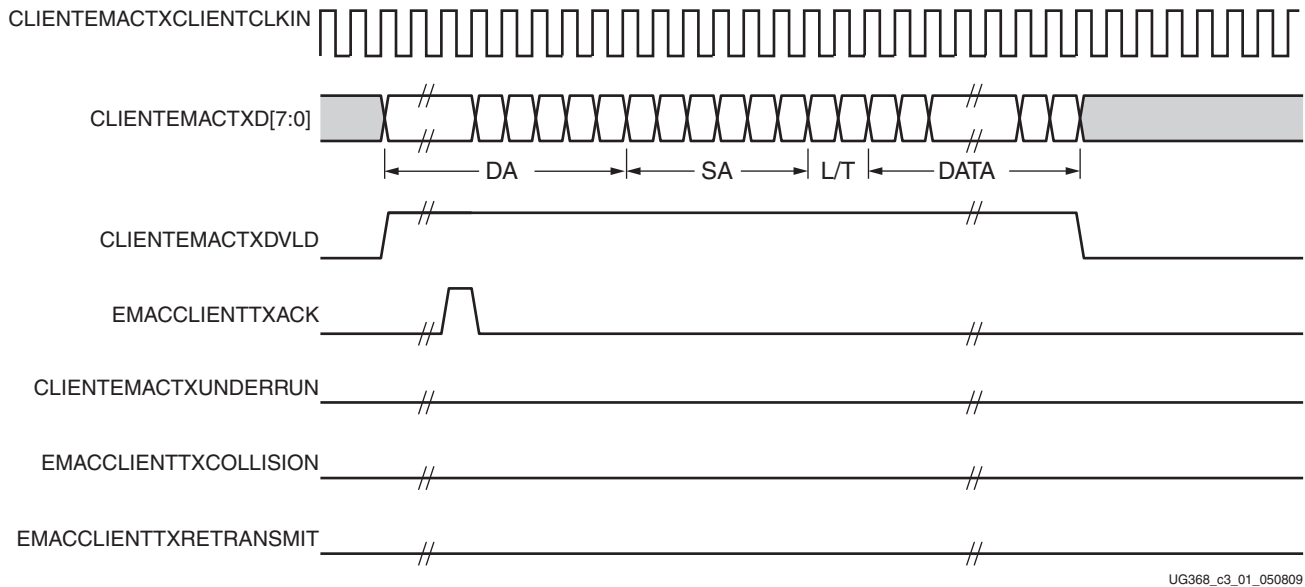


Figure 3-1: Normal Frame Transmission Across Client Interface

## In-Band Parameter Encoding

The Ethernet MAC frame parameters, destination address, source address, length/type, and the FCS are encoded within the same data stream used to transfer the frame payload instead of separate ports. This provides the maximum flexibility in switching applications. The preamble, and optionally the FCS, are added to the frame by the Ethernet MAC.

## Padding

When fewer than 46 bytes of data are supplied by the client to the Ethernet MAC, the transmitter module adds padding – up to the minimum frame length. However, if the Ethernet MAC is configured for client-passed FCS, the client must also supply the padding to maintain the minimum frame length (see [Client-Supplied FCS Passing for Transmission](#), page 47).

## Client-Supplied FCS Passing for Transmission

In the transmission timing case shown in [Figure 3-2](#), an Ethernet MAC is configured to have the FCS field passed in by the client (see [Configuration Registers](#), page 84). The client must ensure that the frame meets the Ethernet minimum frame length requirements. If the frame does not meet these requirements, the Ethernet MAC will pad the frame to the minimum frame length. Although this does not cause the transmitter statistics vector to

indicate a bad frame, it will result in an errored frame as received by the link partner Ethernet MAC.

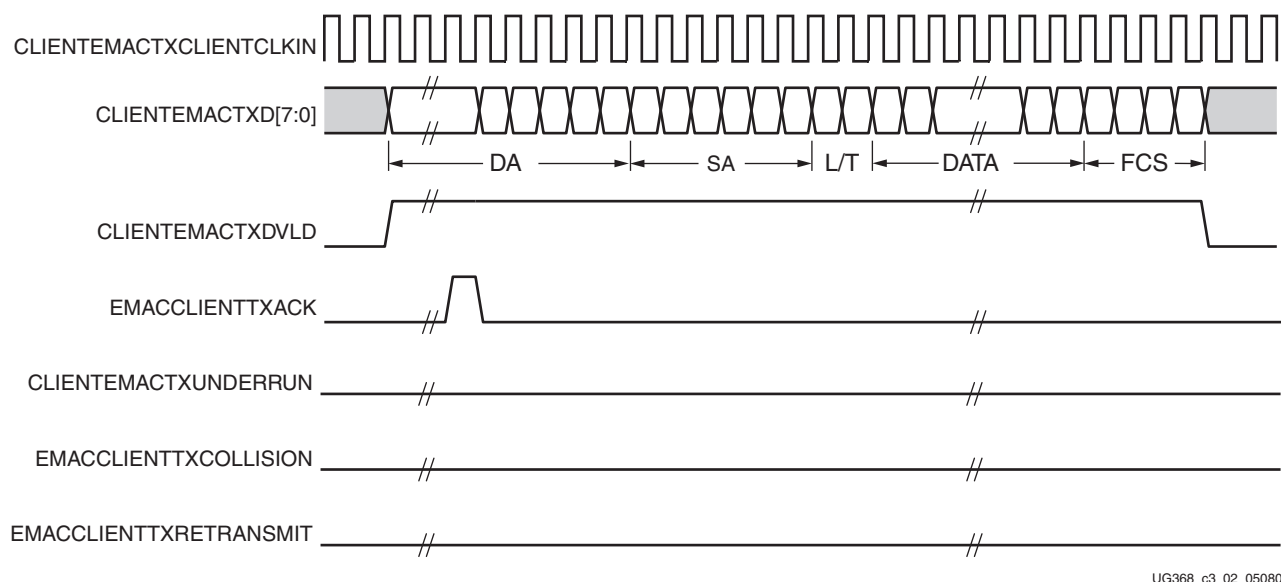


Figure 3-2: Frame Transmission with Client-Supplied FCS

## Client Underrun

The timing of an aborted transfer is shown in Figure 3-3. An aborted transfer can occur if a FIFO connected to the client interface empties before a frame is completed. The CLIENTEMACTXUNDERRUN signal on the client interface is used to signal to the MAC that an underrun condition exists.

When the client asserts CLIENTEMACTXUNDERRUN during a frame transmission, the EMACPHYTXER output is asserted for one clock cycle to notify the external GMII, RGMII, or MII PHY that the frame is corrupted. The client can assert CLIENTTXUNDERRUN at any time during a frame transmission, following the Ethernet MAC's assertion of EMACCLIENTTXACK.

If 1000BASE-X PCS/PMA or SGMII mode is operational, the MAC inserts an error code into the current frame to signal corruption. The Ethernet MAC then falls back to idle transmission. The client must requeue the aborted frame for transmission.

After an underrun occurs, CLIENTEMACTXDVLD must be deasserted for at least one clock cycle before transmission can recommence. CLIENTEMACTXDVLD can be deasserted on the same cycle as underrun is asserted. There is no requirement to deassert CLIENTEMACTXDVLD when underrun is asserted; it can be deasserted later.



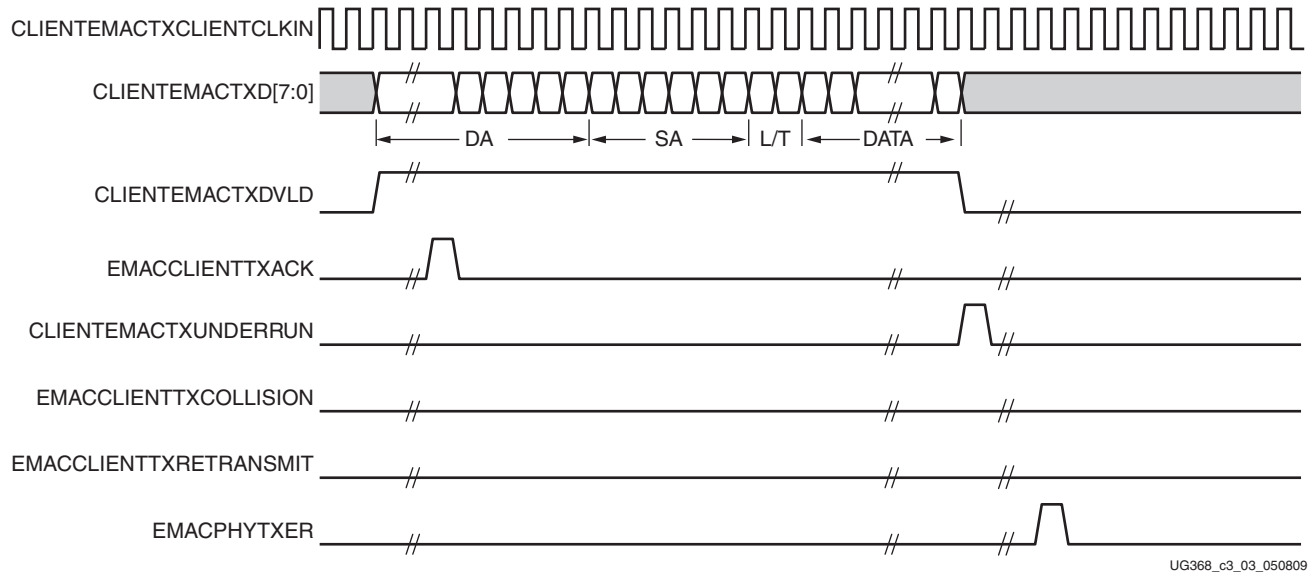


Figure 3-3: Frame Transmission with Underrun

## Back-to-Back Transfers with 8-Bit Client Interface

Back-to-back transfers can occur when the Ethernet MAC client is ready to transmit a second frame of data immediately following completion of the first frame. In Figure 3-4, the end of the first frame is shown on the left. At the clock cycle immediately following the final byte of the first frame, CLIENTEMACTXDVLD is deasserted by the client. One clock cycle later, CLIENTEMACTXDVLD is asserted High, indicating that the first byte of the destination address of the second frame is awaiting transmission on CLIENTEMACTXD.

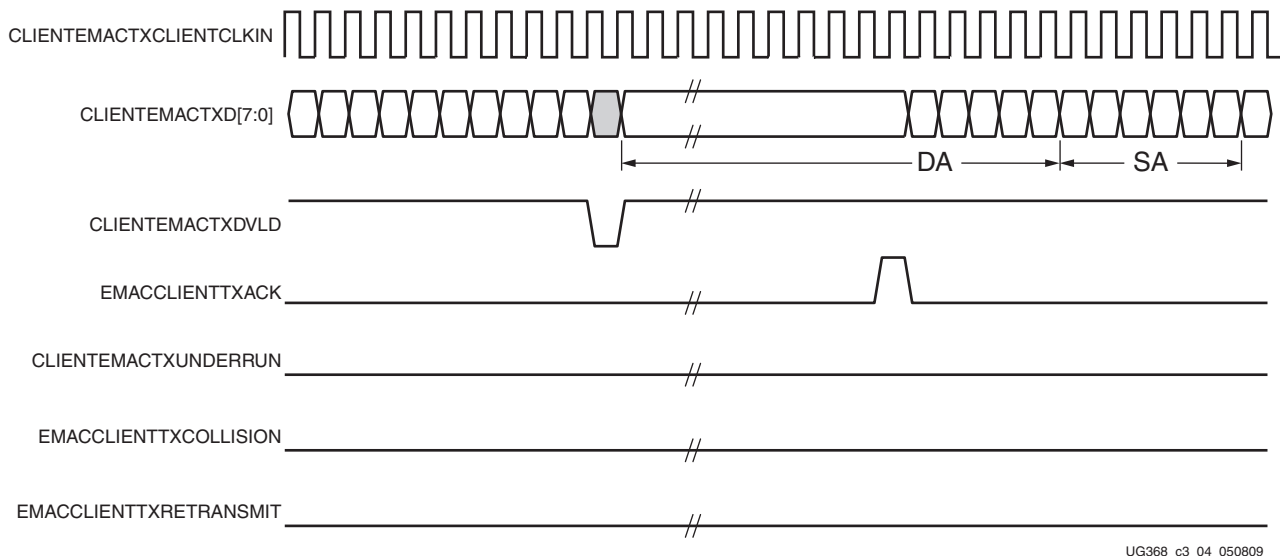


Figure 3-4: Back-to-Back Frame Transmission

When the Ethernet MAC is ready to accept data, EMACCLIENTTXACK is asserted and the transmission continues in the same manner as the single frame case. The Ethernet MAC

defers the assertion of EMACCLIENTTXACK to comply with inter-packet gap requirements and flow control requests.

## Virtual LAN (VLAN) Tagged Frames

Figure 3-5 shows the transmission of a VLAN tagged frame (if enabled). The handshaking signals across the interface do not change. However, the VLAN type tag 0x8100 must be supplied by the client to show the frame as VLAN tagged. The client also supplies the two bytes of tag control information, V1 and V2, at the appropriate positions in the data stream. More information on the contents of these two bytes can be found in the IEEE Std 802.3-2005 specification.

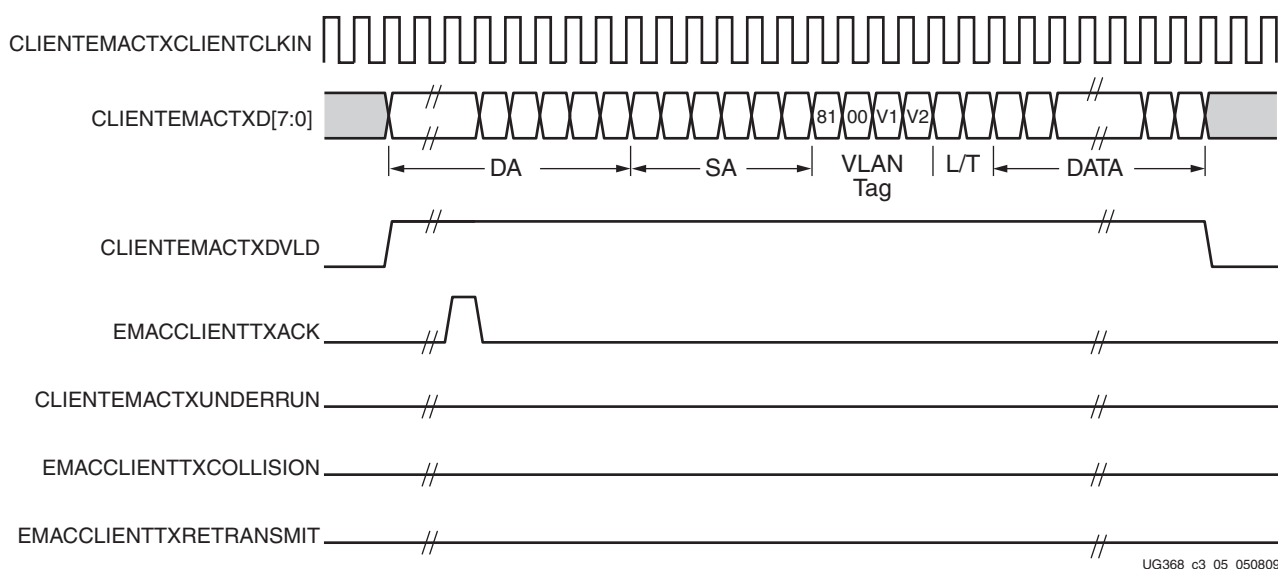


Figure 3-5: Transmission of a VLAN Tagged Frame

## Maximum Permitted Frame Length/Jumbo Frames for Transmission

The maximum length of a frame specified in the IEEE Std 802.3-2005 specification is 1518 bytes for non-VLAN tagged frames. VLAN tagged frames can be extended to 1522 bytes. When jumbo frame handling is disabled and the client attempts to transmit a frame that exceeds the maximum legal length, the Ethernet MAC inserts an error code to corrupt the current frame and the frame is truncated to the maximum legal length. When jumbo frame handling is enabled, frames longer than the legal maximum are transmitted error free.

For more information on enabling and disabling jumbo frame handling, see [Configuration Registers, page 84](#).

## Frame Collisions - Half-Duplex 10/100 Mb/s Operation Only

In half-duplex Ethernet operation, collisions occur on the medium. This is how the arbitration algorithm is fulfilled. When there is a collision, the Ethernet MAC signals to the client a need to have data resupplied as follows:

- If there is a collision, the EMACCLIENTTXCOLLISION signal is set High by the Ethernet MAC. If a frame is in progress, the client must abort the transfer, and CLIENTEMACTXDVLD is deasserted Low.
- If the EMACCLIENTTXRETRANSMIT signal is High in the same clock cycle that the EMACCLIENTTXCOLLISION signal is High, the client must resubmit the previous frame to the Ethernet MAC for retransmission; CLIENTEMACTXDVLD must be asserted to the Ethernet MAC within eight clock cycles of the EMACCLIENTTXCOLLISION signal to meet Ethernet timing requirements. This operation is shown in Figure 3-6. If the frame presented to the client interface is shorter than the collision window (slot time) defined in IEEE Std 802.3-2005, a retransmission request can occur after the end of the frame on the client interface. This can also occur on slightly longer frames due to the latency of the transmit section of the Ethernet MAC. In half duplex mode, the client should wait until the collision window has passed before submitting a new frame.

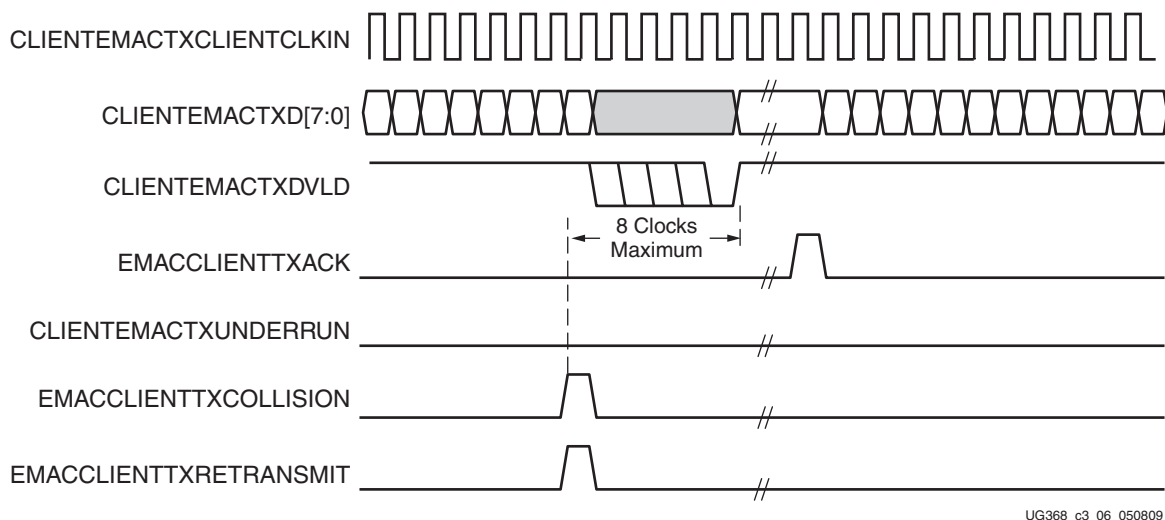


Figure 3-6: Collision Handling - Frame Retransmission Required

- If the EMACCLIENTTXRETRANSMIT signal is Low in the same clock cycle that the EMACCLIENTTXCOLLISION signal is High, the number of retries for this frame has exceeded the Ethernet specification, and the frame should be dropped by the client. The user must drop TXDVLD the cycle after TXCOLLISION is asserted. The client can then make any new frame available to the Ethernet MAC for transmission without timing restriction. This process is shown in Figure 3-7.

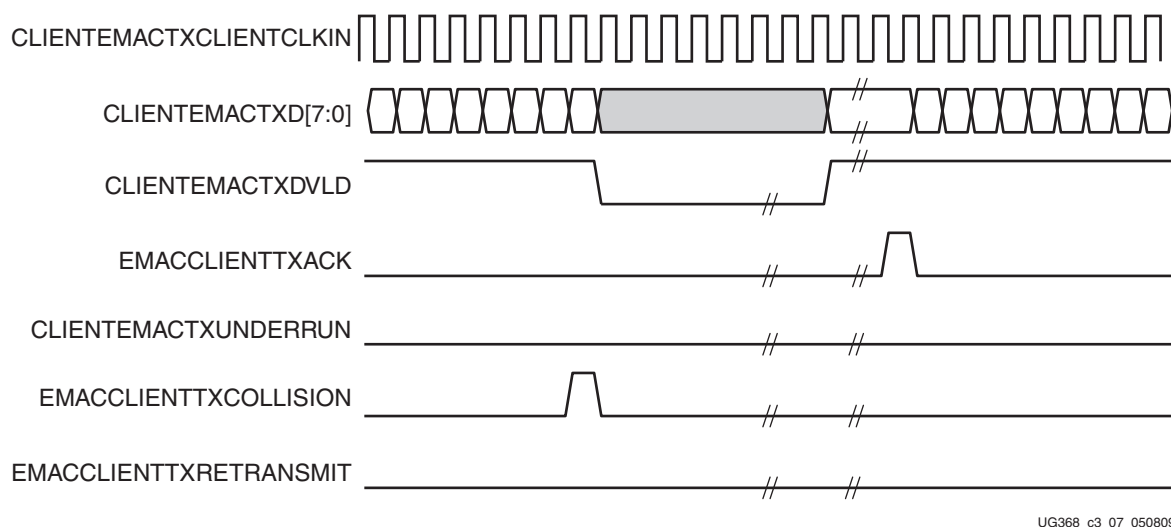


Figure 3-7: Collision Handling - No Frame Retransmission Required

## IFG Adjustment

The length of the IFG can be varied in full-duplex mode. If this function is selected (using a configuration bit in the transmitter control register, see [Configuration Registers, page 84](#)), then the Ethernet MAC exerts back pressure to delay the transmission of the next frame until the requested number of idle cycles has elapsed. The number of idle cycles can be increased. The number of cycles is controlled by the value on the CLIENTEMACTXIFGDELAY port at the start-of-frame transmission if the IFG Adjustment Enable bit in the Transmitter Configuration Register is set. If IFG adjustment is disabled or the value on CLIENTEMACTXIFGDELAY is less than the minimum IFG specified in IEEE Std 802.3-2005, the minimum IFG (12 idles) is used. [Figure 3-8](#) shows the Ethernet MAC operating in this mode.

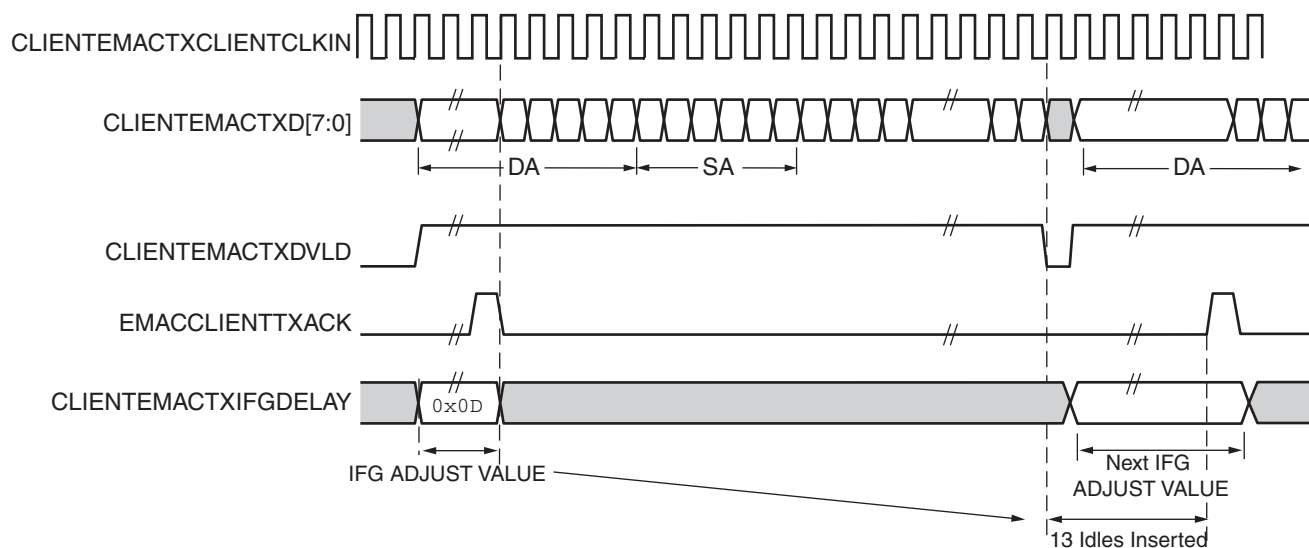


Figure 3-8: IFG Adjustment

## Transmit (TX) Client: 8-Bit Interface (with Clock Enables)

This section covers the client transmit interface when it is configured to be 8 bits wide and when the optional clock enable clocking scheme is used. This optional configuration can be used when the Ethernet MAC is configured in MII/GMII or RGMII mode. [Ethernet MAC Attributes, page 36](#) describes the EMAC\_USECLKEN attribute, which is set to TRUE for this configuration.

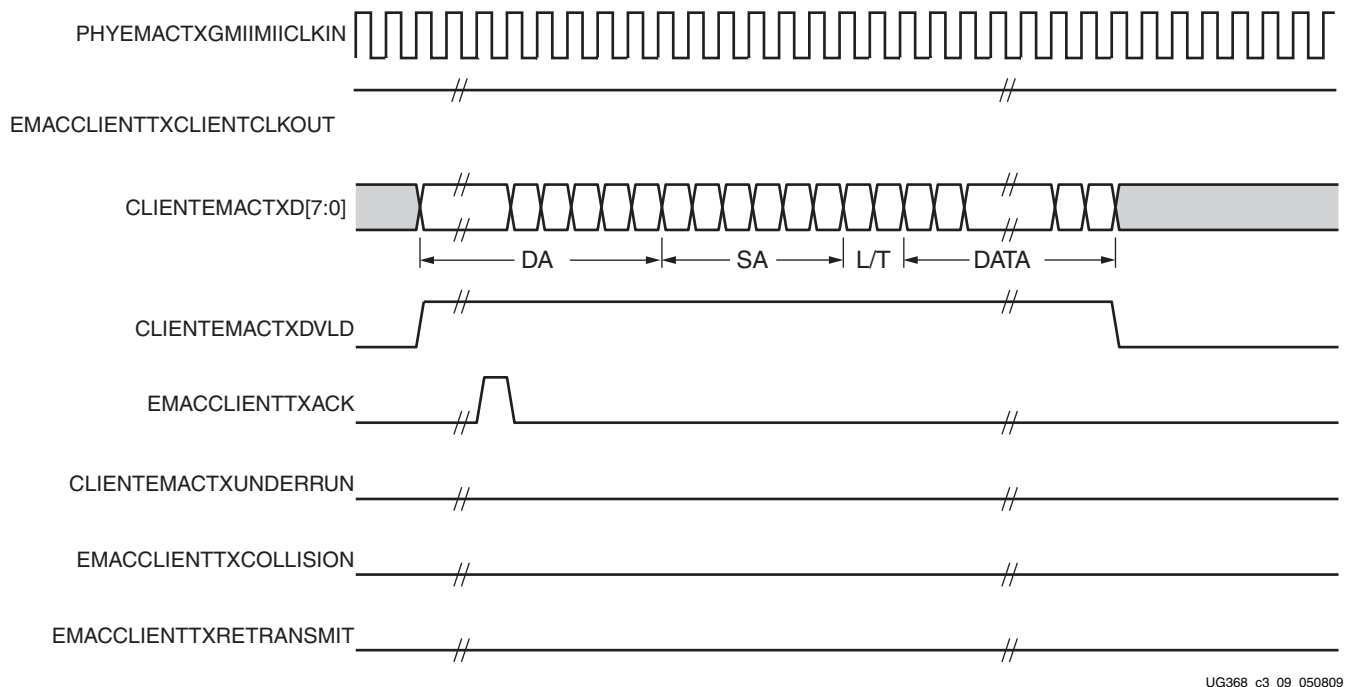
In this mode, the client interface is synchronous to the physical interface clock. This configuration reduces the number of BUFs that are used to implement the Ethernet MAC design. See [Ethernet MAC Clocks, page 193](#) for some general information on this clocking scheme. More details are available for specific physical interface configurations in the following sections:

- [MII Clock Management Using Clock Enables](#)
- [GMII Clock Management for Tri-Speed Operation Using Clock Enables](#)
- [RGMII Clock Management for Tri-Speed Operation Using Clock Enables](#)

For the transmit interface, the PHYEMACGMIIIMCLKIN signal is used as the clock and the EMACCLIENTTXCLIENTCLKOUT signal is used as the clock enable.

### Normal Frame Transmission with 8-Bit Client Interface and Clock Enables

[Figure 3-9](#) shows the transmission of a normal frame at 1 Gb/s. At 1 Gb/s, no clock enable is required, and the clock enable signal EMACCLIENTTXCLIENTCLKOUT is held High. The data is input on the rising edge of the 125 MHz PHYEMACGMIIIMCLKIN clock.



UG368\_c3\_09\_050809

**Figure 3-9: Normal Frame Transmission at 1 Gb/s with EMAC\_USECLKEN Set to TRUE**

At speeds slower than 1 Gb/s, the physical interface clock runs twice as fast as the client interface clock. The data is still synchronous to PHYEMACGMIIIMCLKIN, but to maintain the correct data rate, the EMACCLIENTTXCLIENTCLKOUT clock signal is used

as an enable rather than a clock. The EMACCLIENTXCLIENTCLKOUT signal is asserted every second cycle of PHYEMACTXGMIIMICLKIN. Data is presented to the transmitter interface only when EMACCLIENTXCLIENTCLKOUT is High. This is illustrated in Figure 3-10.

**Note:** Due to the timing relationship between the internal and external clocks, the EMACCLIENTXACK signal should be registered at the output of the EMAC at speed lower than 1 Gb/s.

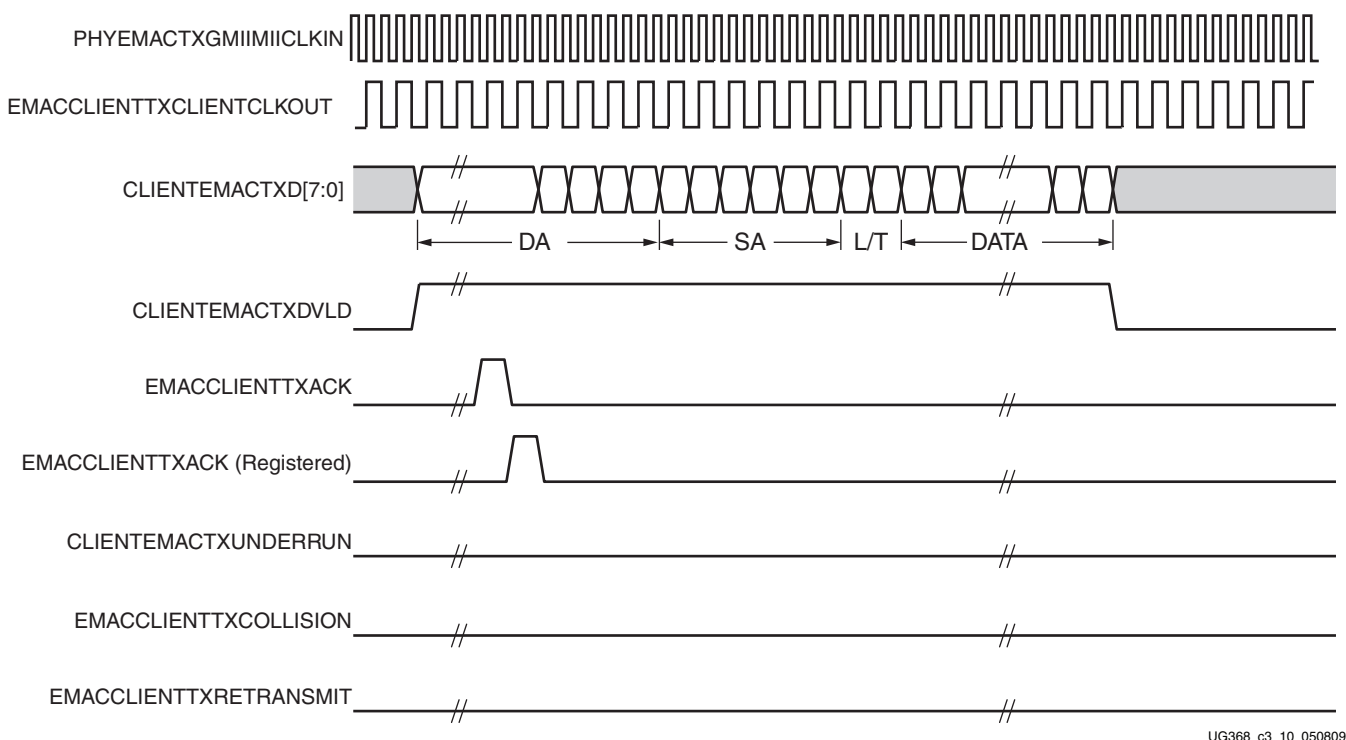


Figure 3-10: Normal Frame Transmission at 10/100 Mb/s with EMAC\_USECLKEN Set to TRUE

## Transmit (TX) Client: 16-Bit Interface

This optional configuration can only be used when the Ethernet MAC physical interface is configured in 1000BASE-X PCS/PMA mode.

When a 16-bit client interface is selected, the input clock for the Ethernet MAC TX client logic (CLIENTEMACTXCLIENTCLKIN) should be twice the frequency of the clock used for the TX client logic in the FPGA logic. Figure 6-19, page 161 illustrates how this clocking scheme is achieved for this mode of operation.

This mode can operate with the Ethernet MAC TX client logic clocked at 125 MHz and the logic in the FPGA logic clocked at 62.5 MHz. However, this mode allows the Ethernet MAC TX client logic to operate at up to 312.5 MHz while the logic in the FPGA logic is clocked at 156.25 MHz. By using the faster clock speeds, a data rate of up to 2.5 Gb/s can be achieved. This is greater than 1 Gb/s, as specified in IEEE Std 802.3-2005. Therefore, this interface should not be used for Ethernet-compliant designs, but it can be useful for backplane applications.

Ethernet MAC Attributes, page 36 describes the EMAC\_TX16BITCLIENT\_ENABLE attribute, which is set to TRUE for this configuration.

Figure 3-11 shows the timing of a normal outbound frame transfer for the case with an even number of bytes in the frame. The CLIENTEMACTXCLIENTCLKIN clock is used as an input to the Ethernet MAC and the PHYEMACMIITXCLK clock (CLIENTEMACTXCLIENTCLKIN/2) is used to clock the TX client logic in the FPGA logic.

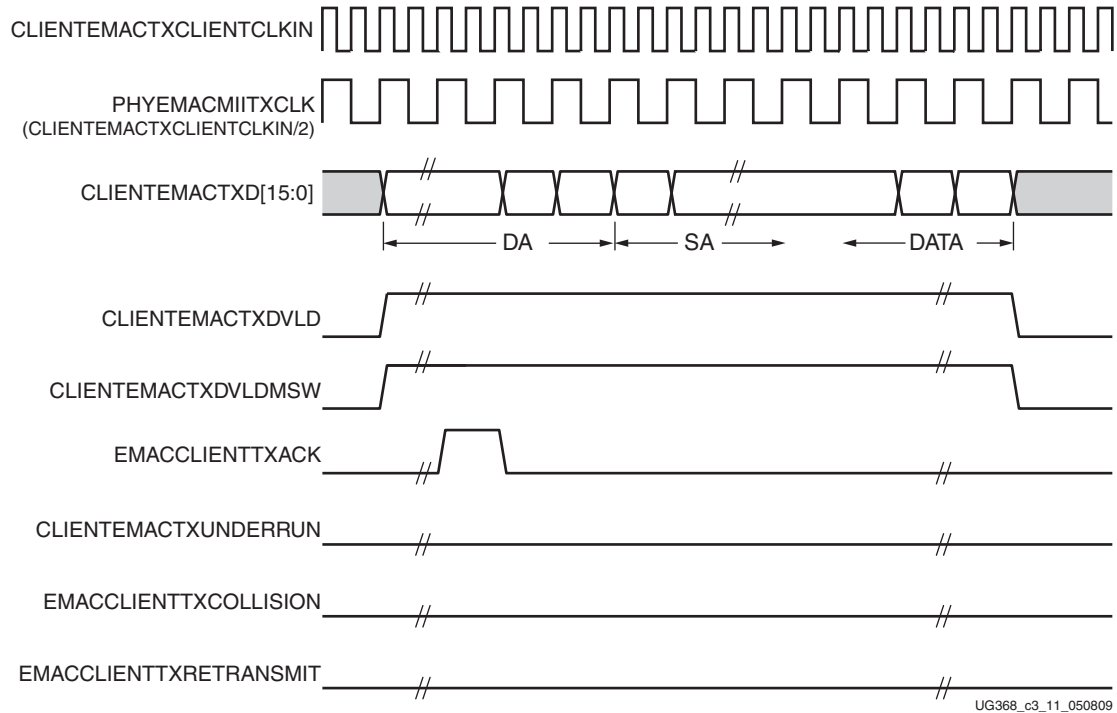


Figure 3-11: 16-Bit Transmit (Even Byte Case)

Figure 3-12 shows the timing of a normal outbound frame transfer for the case with an odd number of bytes in the frame.

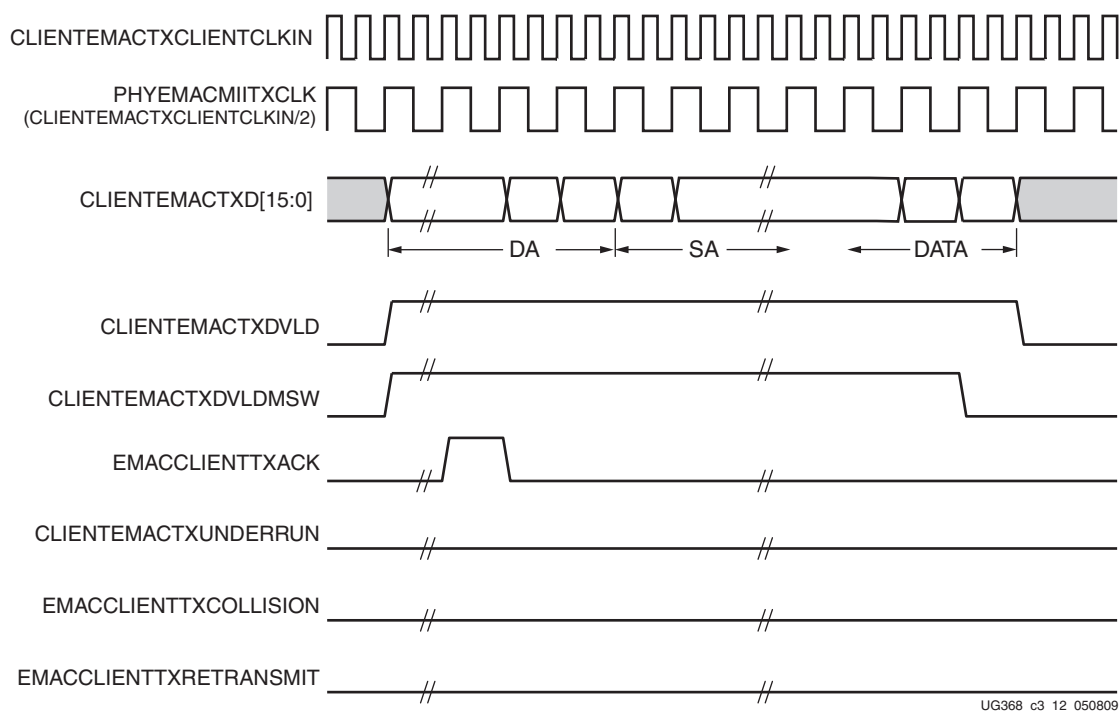


Figure 3-12: 16-Bit Transmit (Odd Byte Case)

As shown in Figure 3-12, CLIENTEMACTXDVLDMSW denotes an odd number of bytes in the frame. In the odd byte case, CLIENTEMACTXDVLDMSW is deasserted one clock cycle earlier than the CLIENTEMACTXDVLD signal, after the transmission of the frame. Otherwise, these data valid signals are the same as shown in the even byte case (Figure 3-11).

## Back-to-Back Transfers with 16-Bit Client Interface

For back-to-back transfers, both CLIENTEMACTXDVLD and CLIENTEMACTXDVLDMSW must be deasserted for one PHYEMACMIITXCLK (half the clock frequency of CLIENTEMACTXCLIENTCLKIN) clock cycle after the first frame. In the following PHYEMACMIITXCLK clock cycle, both CLIENTEMACTXDVLD and CLIENTEMACTXDVLDMSW are asserted High to indicate that the first two bytes of the destination address of the second frame is ready for transmission on CLIENTEMACTXD[15:0]. In 16-bit mode, this one PHYEMACMIITXCLK clock cycle IFG corresponds to a 2-byte gap (versus a 1-byte gap in 8-bit mode) between frames in the back-to-back transfer.

Figure 3-13 shows the timing diagram for 16-bit transmit for an even-byte case, and Figure 3-14 shows the timing diagram for an odd-byte case.



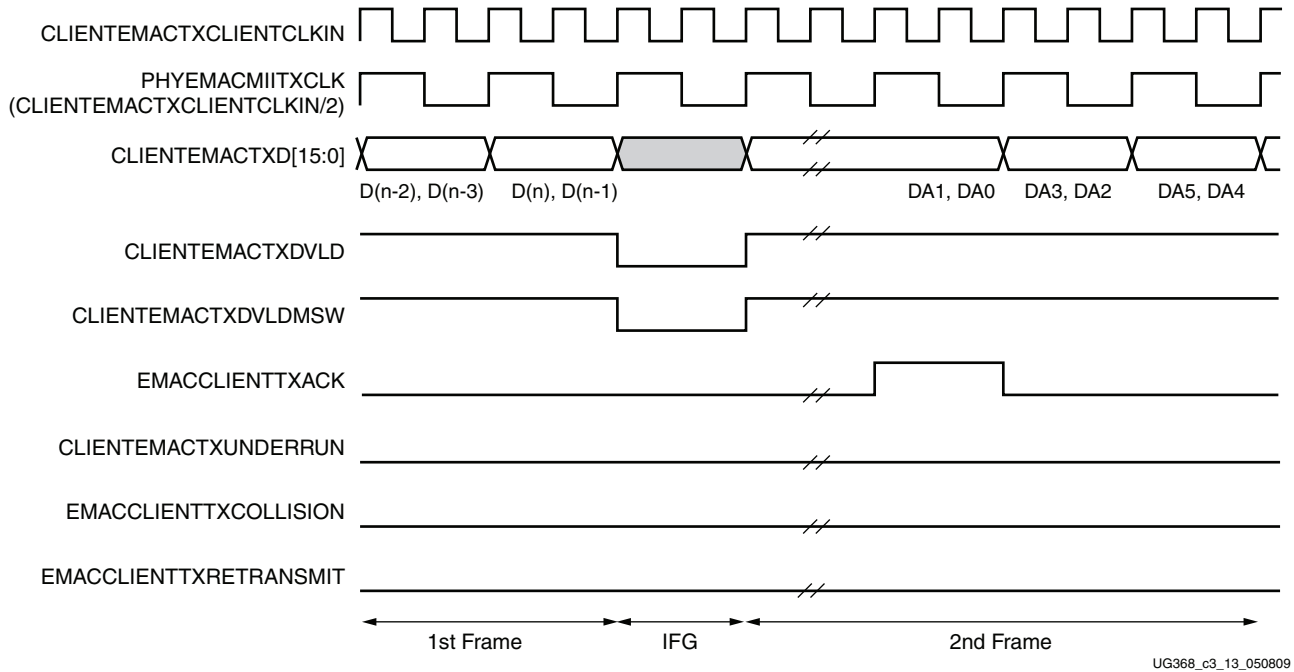


Figure 3-13: 16-Bit Transmit Back-to-Back Transfer (Even Byte Case)

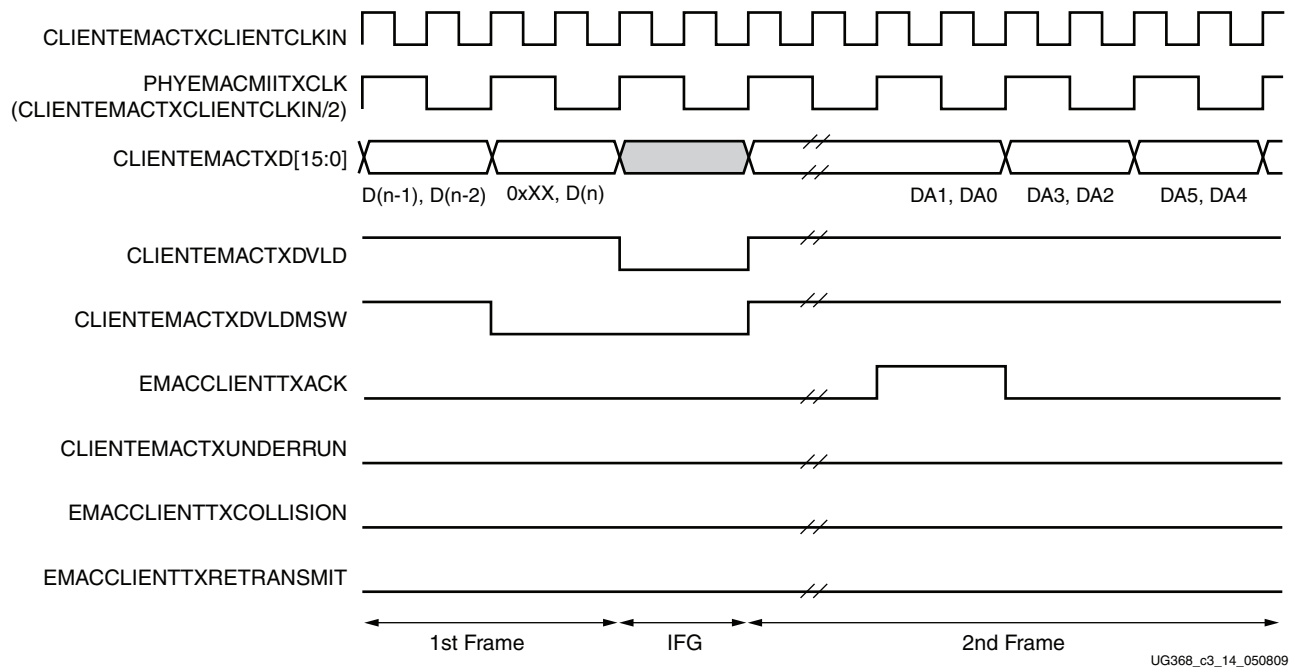


Figure 3-14: 16-Bit Transmit Back-to-Back Transfer (Odd Byte Case)

## Receive (RX) Client: 8-Bit Interface (without Clock Enables)

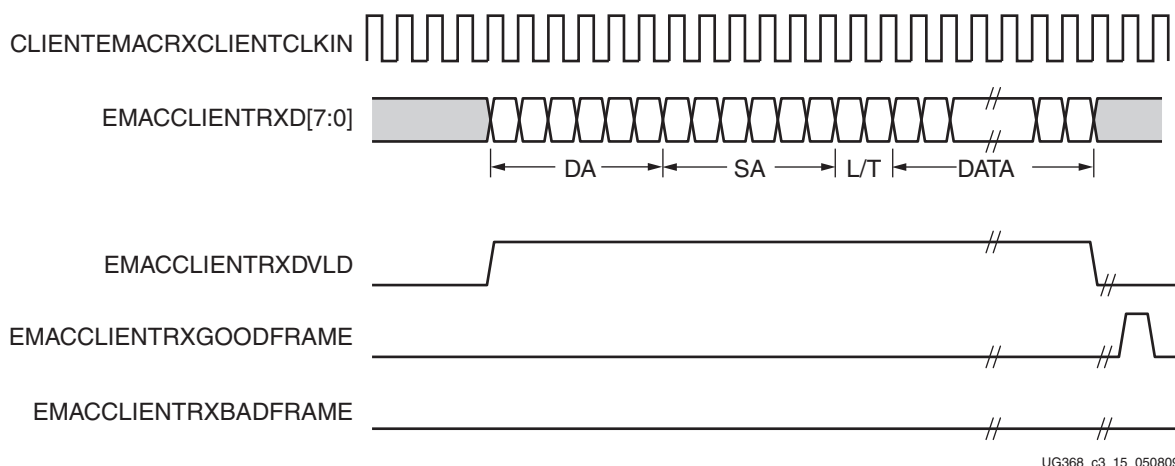
This section covers the client receive interface when it is configured to be 8 bits wide and when the default clocking scheme is used. [Ethernet MAC Attributes, page 36](#) describes the EMAC\_USECLKEN attribute, which is set to FALSE for this configuration. In this configuration, EMACCLIENTRXD[15:8] and EMACCLIENTRXDVLDMSW are left unconnected because the upper eight bits of the data bus are not used.

See [Ethernet MAC Clocks, page 193](#) for some general information on this standard clocking scheme. More details are available for specific physical interface configurations in the following sections:

- MII
  - [MII Standard Clock Management](#)
- GMII
  - [GMII Clock Management for 1 Gb/s Only](#)
  - [GMII Standard Clock Management for Tri-Speed Operation](#)
- RGMII
  - [RGMII Clock Management for 1 Gb/s Only](#)
  - [RGMII Standard Clock Management for Tri-Speed Operation](#)
- SGMII
  - [SGMII Clock Management](#)
- 1000BASE-X
  - [1000BASE-X PCS/PMA Clock Management](#)

### Normal Frame Reception

The timing of a normal inbound frame transfer is shown in [Figure 3-15](#). The client must accept data at any time; there is no buffering within the Ethernet MAC to allow for receive client latency. After frame reception begins, data is transferred on consecutive clock cycles to the receive client until the frame is complete. The Ethernet MAC asserts the EMACCLIENTRXGOODFRAME signal to indicate successful receipt of the frame and the ability to analyze the frame by the client.



UG368\_c3\_15\_050809

Figure 3-15: Normal Frame Reception

Frame parameters (destination address, source address, LT, data, and optionally FCS) are supplied on the data bus as shown in the timing diagram. The abbreviations are the same as those described in [Table 3-1, page 46](#).

If the LT field has a length interpretation, the inbound frame could be padded to meet the Ethernet minimum frame size specification. This padding is not passed to the client in the data payload; an exception is when FCS passing is enabled. See [Client-Supplied FCS Passing for Reception, page 60](#).

Therefore, when client-supplied FCS passing is disabled, EMACCLIENTRXDVLD is Low between frames for the duration of the padding field (if present), the FCS field, carrier extension (if present), the IFG following the frame, and the preamble field of the next frame. When client-supplied FCS passing is enabled, EMACCLIENTRXDVLD is Low between frames for the duration of carrier extension (if present), the IFG, and the preamble field of the following frame.

## Signal Timing of Good/Bad Frame Pulses

Although the timing diagram in [Figure 3-15](#) shows the EMACCLIENTRXGOODFRAME signal asserted shortly after the last valid data on EMACCLIENTRXD[7:0], this is not always the case. The EMACCLIENTRXGOODFRAME or EMACCLIENTRXBADFRAME signals are asserted only after completing all the frame checks. These frame checks occur only after receipt of any padding (if present), followed by the FCS and any carrier extension (if present). Therefore, either EMACCLIENTRXGOODFRAME or EMACCLIENTRXBADFRAME is asserted following frame reception at the beginning of the IFG.

## Frame Reception with Errors

An unsuccessful frame reception (for example, a fragment frame or a frame with an incorrect FCS) is shown in [Figure 3-16](#). In this case, the EMACCLIENTRXBADFRAME signal is asserted to the client after the end of the frame. The client is responsible for dropping the data already transferred for this frame.

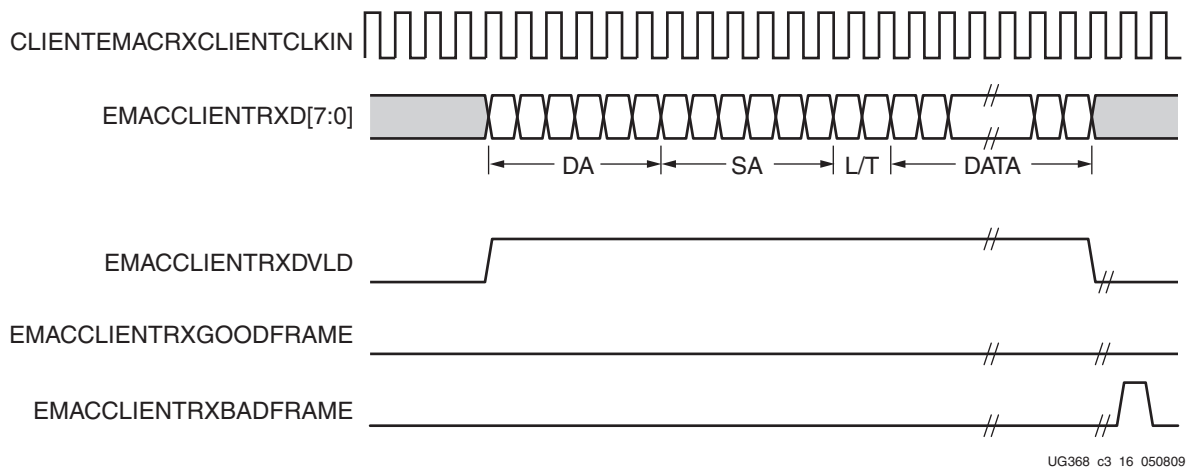


Figure 3-16: Frame Reception with Error

The following conditions cause the assertion of BAD\_FRAME.

## Standard Conditions

- FCS errors occur.
- Packets are shorter than 64 bytes (undersize or fragment frames).
- Jumbo frames are received when jumbo frames are not enabled.
- The length/type field is length, but the real length of the received frame does not match the value in the length/type field (when length/type checking is enabled).
- The length/type field is length, in which the length value is less than 46. In this situation, the frame should be padded to minimum length. If it is not padded to exactly minimum frame length, the frame is marked as bad (when length/type checking is enabled).
- Any control frame that is received is not exactly the minimum frame length (when control frame length checking is enabled).
- PHYEMACRXER is asserted at any point during frame reception.
- An error code is received in the 1 Gb frame extension field.
- A valid pause frame, addressed to the MAC, is received when flow control is enabled. Please see [Flow Control Block](#) for more information.

## 1000BASE-X/SGMII Specific Conditions

When in 1000BASE-X mode or SGMII mode, the following errors can also cause a frame to be marked as bad:

- Unrecognized 8B/10B code group received during the packet.
- 8B/10B running disparity errors occur during the packet.
- Unexpected K characters or sequences appearing in the wrong order/byte position.

## Client-Supplied FCS Passing for Reception

[Figure 3-17](#) shows how the Ethernet MAC is configured to pass the FCS field to the client (see [Configuration Registers](#), page 84). In this case, any padding inserted into the frame to meet Ethernet minimum frame length specifications is left intact and passed to the client. Even though the FCS is passed to the client, it is also verified by the Ethernet MAC, and EMACCLIENTRXBADFRAME is asserted if the FCS check fails.

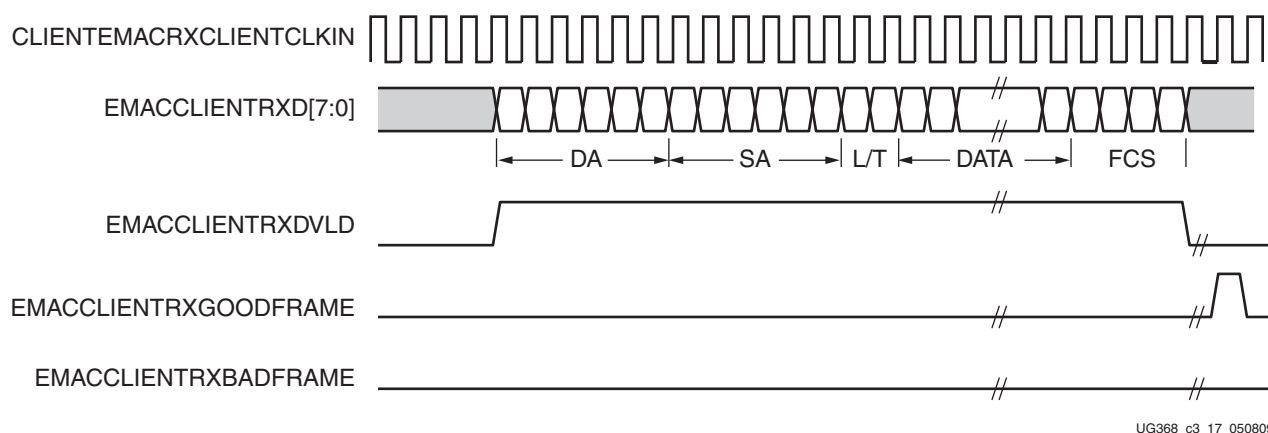


Figure 3-17: Frame Reception with In-Band FCS Field

## VLAN Tagged Frames

The reception of a VLAN tagged frame (if enabled) is shown in Figure 3-18. The VLAN frame is passed to the client to identify the frame as VLAN tagged. This is followed by the tag control information bytes, V1 and V2. The length/type field after the tag control information bytes is not checked for errors. More information on the interpretation of these bytes is described in the IEEE Std 802.3-2005 standard.

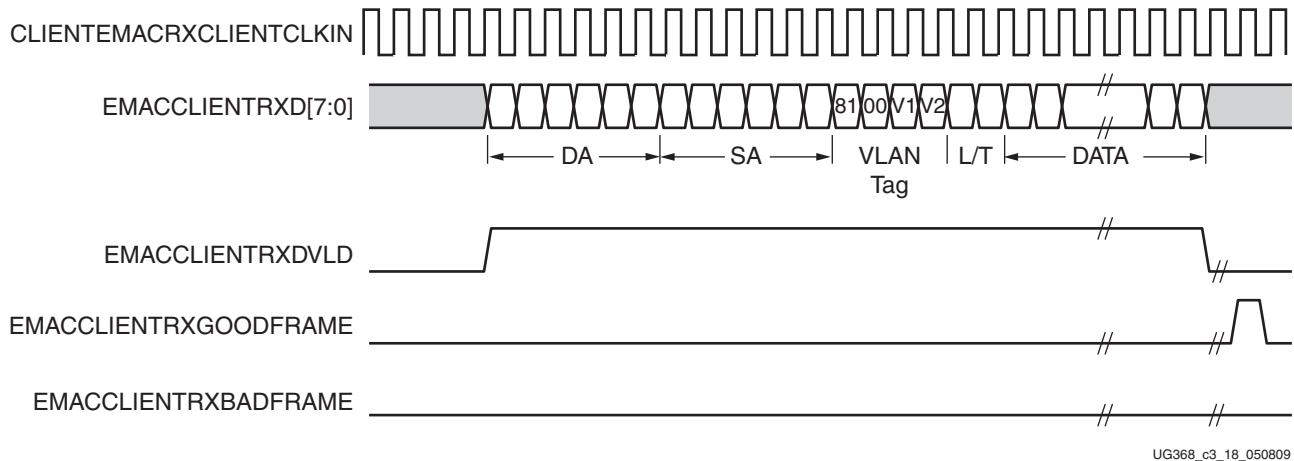


Figure 3-18: Reception of a VLAN Tagged Frame

## Maximum Permitted Frame Length/Jumbo Frames for Reception

The maximum length of a frame specified in IEEE Std 802.3-2005 is 1518 bytes for non-VLAN tagged frames. VLAN tagged frames can be extended to 1522 bytes. When jumbo frame handling is disabled and the Ethernet MAC receives a frame exceeding the maximum legal length, EMACCLIENTRXBADFRAME is asserted. When jumbo frame handling is enabled, frames longer than the legal maximum are received in the same way as shorter frames.

For more information on enabling and disabling jumbo frame handling, see Table 4-3, page 84.

## Length/Type Field Error Checks

Length/Type Field Error checking is specified in IEEE Std 802.3-2005. This functionality must be enabled to comply with this specification. Disabling Length/Type checking is intended only for specific applications, such as when using over a proprietary backplane.

### Enabled

When length/type error checking is enabled (see Table 4-3, page 84), the following checks are made on all frames received. (If either of these checks fails, EMACCLIENTRXBADFRAME is asserted):

- A value greater than or equal to decimal 46 but less than decimal 1500 (a length interpretation) in the length/type field is checked against the actual data length received.

- A value less than decimal 46 in the length/type field is checked to ensure the data field is padded to exactly 46 bytes. The resultant frame is now the minimum frame size of 64 bytes total in length.

Furthermore, if padding is indicated (the length/type field is less than decimal 46) and client-supplied FCS passing is disabled, then the length value in the length/type field is used to deassert EMACCLIENTRXDVLD after the indicated number of data bytes removing the padding bytes from the frame.

When a frame is a VLAN frame, only the initial length/type field containing the VLAN type  $0 \times 8100$  is checked and interpreted as a type. The subsequent length/type field in the VLAN frame is not checked.

## Disabled

When the length/type error checking is disabled (see [Table 4-3, page 84](#)), the length/type error checks previously described are not performed. A frame containing only these errors causes EMACCLIENTRXGOODFRAME to be asserted.

Furthermore, if padding is indicated and client-supplied FCS passing is disabled, then a length value in the length/type field is not used to deassert EMACCLIENTRXDVLD. Instead EMACCLIENTRXDVLD is deasserted before the start of the FCS field; in this way, any padding is not removed from the frame.

It should be noted that an illegal length frame, e.g., a frame of less than 64 bytes in total length, is still marked as bad. In addition, the disabling of Length/Type checks has no effect on control frames. If a control frame is received that is not 64 bytes in total length, it is still marked as a bad frame unless control frame length checking is disabled.

## Control Frame Length Checks

When control frame length checking is enabled (see [Table 4-3, page 84](#)), the Ethernet MAC accepts control frames, designated by a Length/Type field containing the value  $0 \times 8808$ , of exactly minimum frame length only. Frames of other lengths are discarded.

When control frame length checking is disabled, the Ethernet MAC receiver does not enforce this requirement, and instead ensures only that the frame is of legal length; namely, that the frame is at least 64 bytes in total length, and not greater than 1518 bytes in total length. When both control frame length checking is disabled and jumbo frame reception is enabled, control frames can exceed 1518 total bytes.

The IEEE Std 802.3-2005 standard permits the operation exhibited by the Ethernet MAC, both when control frame length checking is enabled and when it is disabled.

## Receive (RX) Client: 8-Bit Interface (with Clock Enables)

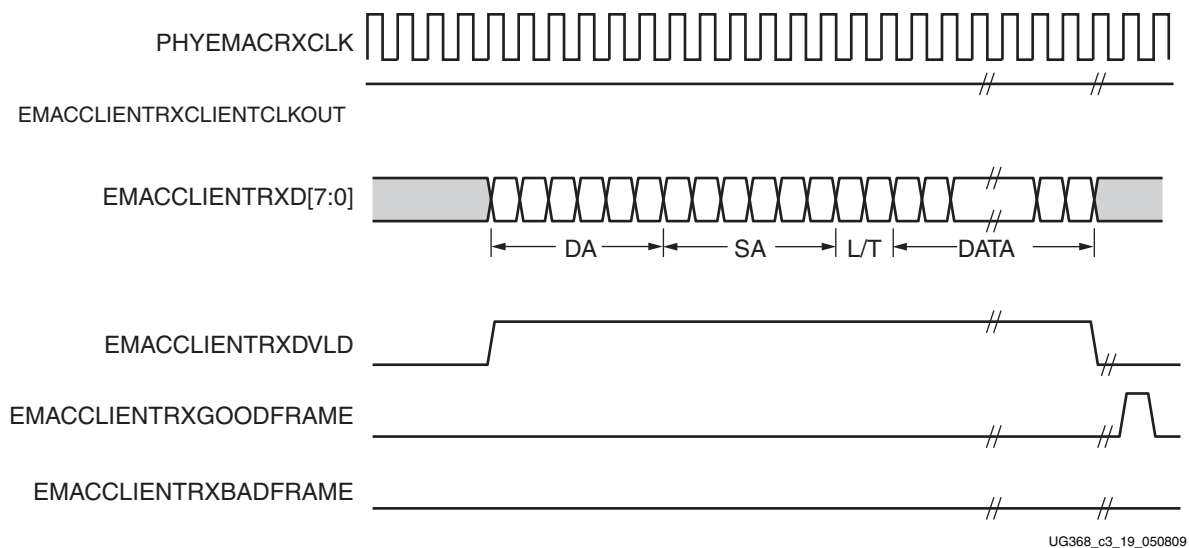
This section covers the client receive interface when it is configured to be 8 bits wide and when the optional clock enable clocking scheme is used. This optional configuration can be used when the Ethernet MAC is configured in MII/GMII or RGMII mode. [Ethernet MAC Attributes, page 36](#) describes the EMAC\_USECLKEN attribute, which is set to TRUE for this configuration.

In this mode, the client interface is synchronous to the physical interface clock. This configuration reduces the number of BUFs that are used to implement the Ethernet MAC design. See [Ethernet MAC Clocks, page 193](#) for some general information on this clocking scheme. More details are available for specific physical interface configurations in the following sections:

- [MII Clock Management Using Clock Enables](#)
- [GMII Clock Management for Tri-Speed Operation Using Clock Enables](#)
- [RGMII Clock Management for Tri-Speed Operation Using Clock Enables](#)

For the receive interface, the PHYEMACRXCLK signal is used as the clock, and the EMACCLIENTRXCLIENTCLKOUT signal is used as the clock enable.

[Figure 3-19](#) shows the timing of a normal inbound frame transfer at 1 Gb/s. Data is transferred to the client on the rising edge of PHYEMACRXCLK. The output clock enable, EMACCLIENTRXCLIENTCLKOUT, is held High.



**Figure 3-19: Normal Frame Reception at 1 Gb/s with EMAC\_USECLKEN Set to TRUE**

[Figure 3-20](#) shows the reception of a frame at speeds below 1 Gb/s. The physical interface clock runs at twice the speed of the client interface clock and so the output clock enable, EMACCLIENTRXCLIENTCLKOUT, is only asserted on every second cycle of PHYEMACRXCLK. Data is written into the client on the rising edge of PHYEMACRXCLK when EMACCLIENTRXCLIENTCLKOUT is High.

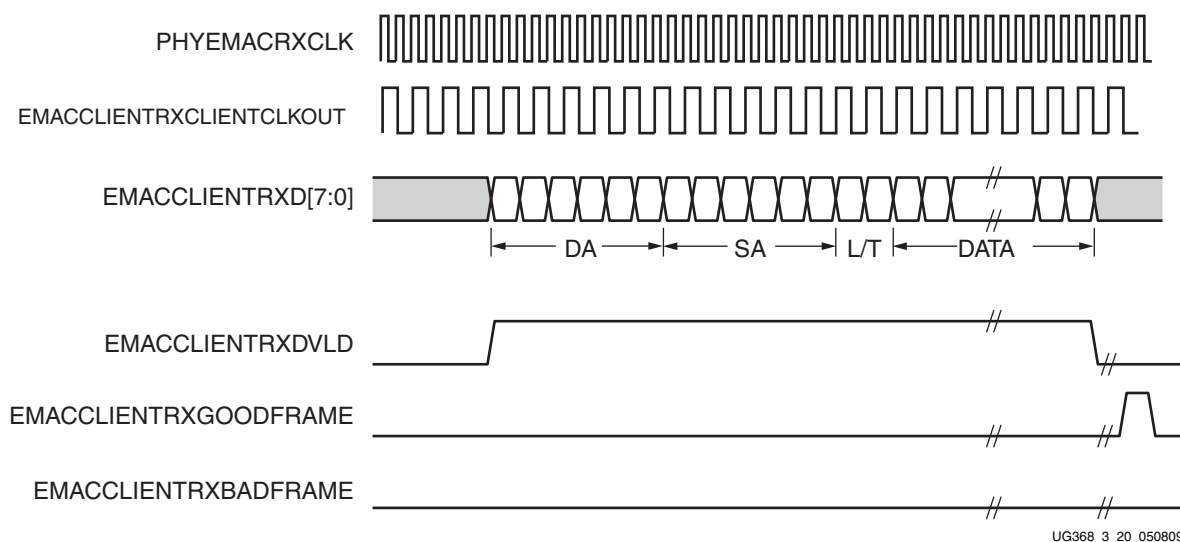


Figure 3-20: Normal Frame Reception at 10/100 Mb/s with EMAC\_USECLKEN Set to TRUE

## Receive (RX) Client: 16-Bit Interface

This optional configuration can only be used when the Ethernet MAC physical interface is configured in 1000BASE-X PCS/PMA mode.

When a 16-bit client interface is selected, the input clock for the Ethernet MAC RX client logic (CLIENTEMACRXCLIENTCLKIN) should be twice the frequency of the clock used for the RX client logic in the FPGA logic. [Figure 6-20, page 163](#) illustrates how this clocking scheme is achieved for this mode of operation.

This mode can operate with the Ethernet MAC RX client logic clocked at 125 MHz and the logic in the FPGA logic clocked at 62.5 MHz. However, this mode allows the Ethernet MAC RX client logic to operate at up to 312.5 MHz while the logic in the FPGA logic is clocked at 156.25 MHz. By using the faster clock speeds, a data rate of up to 2.5 Gb/s can be achieved. This is greater than 1 Gb/s as specified in IEEE Std 802.3-2005. Therefore, this interface should not be used for Ethernet-compliant designs, but it can be useful for backplane applications.

[Ethernet MAC Attributes, page 36](#) describes the EMAC\_RX16BITCLIENT\_ENABLE attribute, which is set to TRUE for this configuration.

[Figure 3-21](#) shows the timing of a normal inbound frame transfer for the case with an even number of bytes in the frame. The CLIENTEMACRXCLIENTCLKIN clock is used as an input to the Ethernet MAC and the PHYEMACRXCLK clock (CLIENTEMACRXCLIENTCLKIN/2) is used to clock the RX client logic in the FPGA logic.



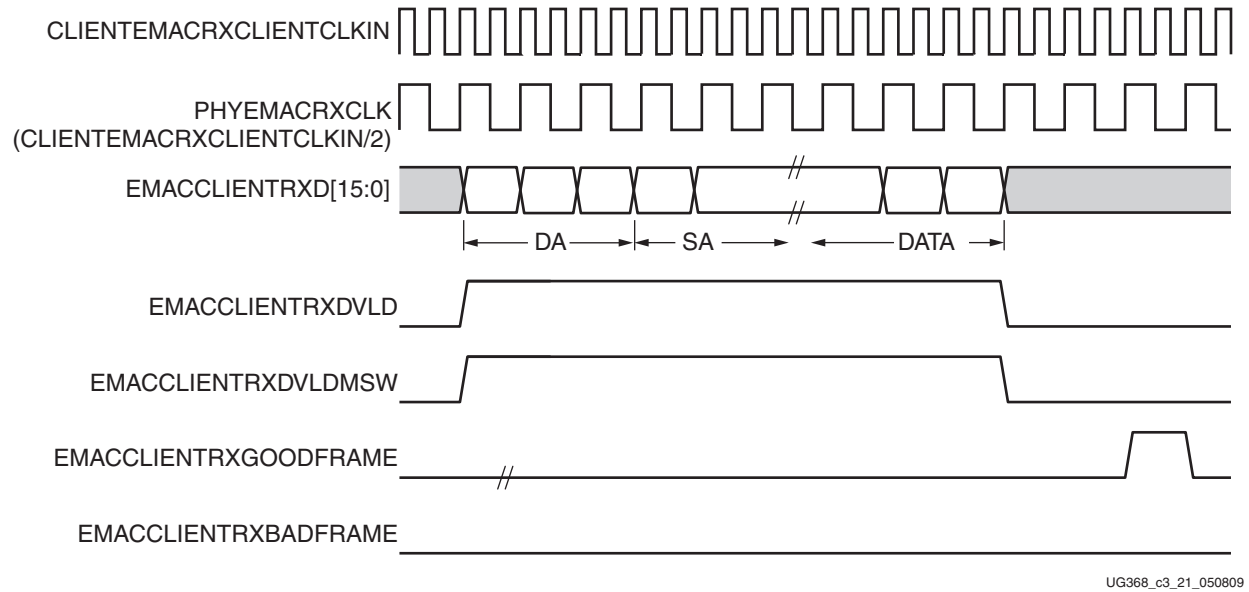


Figure 3-21: 16-Bit Receive (Even Byte Case)

Figure 3-22 shows the timing of a normal inbound frame transfer for the case with an odd number of bytes in the frame.

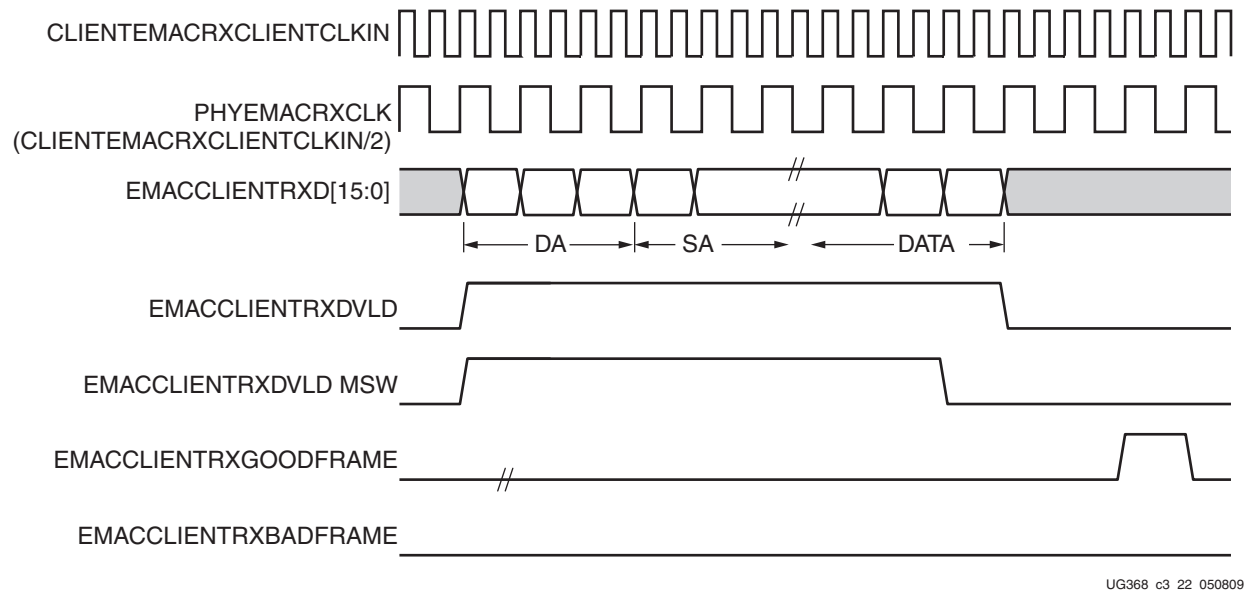


Figure 3-22: 16-Bit Receive (Odd Byte Case)

As shown in Figure 3-21 and Figure 3-22, EMACCLIENTRXDVLDMSW is used to denote an odd number of bytes in the frame. The data valid signals are shown in the even byte case (Figure 3-21). In the odd byte case (Figure 3-22), EMACCLIENTRXDVLDMSW is deasserted one clock cycle earlier compared to the EMACCLIENTRXDVLD signal, after the reception of the frame. EMACCLIENTRXD[7:0] contains the data in this odd byte case.

## Address Filtering

The address filtering block accepts or rejects frames by examining the destination address of an incoming frame. This block includes:

- Matching of programmable unicast destination address
- Matching of four additional programmable general addresses
- Broadcast address recognition (0xFFFF\_FFFF\_FFFF)
- Optional pass-through mode with address filter disabled (promiscuous mode)
- PAUSE control frame address recognition (0x0100\_00C2\_8001)

The Address Filter (AF) protects the client from extraneous traffic. With this technique, the hardware matches the Destination Address (DA) field of the Ethernet MAC frame. This relieves the task from the bus and software.

The AF is programmed in software through the host interface. PAUSE frame addresses and broadcast address are hardwired; they do not need to be programmed. The AF can be enabled and disabled under software control, using an enable bit in the control register. See [Address Filter Registers, page 89](#) for the control register.

When the function is enabled, Ethernet frames are passed to the client interface only if they pass the filter. When the AF function is disabled, all incoming RX frames are passed to the client interface.

For system monitoring, the event of a frame failing the filter is signaled. Equally, when a frame passes the filter, a match is indicated to the client by using the output pins EMACCLIENTRXDVLD and EMACCLIENTRXFRAMEDROP together. [Table 3-2](#) shows the values of the two signals for possible outcomes of an incoming RX frame when the AF is enabled.

When the AF is enabled, it is impossible to determine if there is an incoming RX frame or if the AF has rejected incoming RX frames (using only EMACCLIENTRXDVLD) because in both cases, EMACCLIENTRXDVLD is deasserted. However, using the EMACCLIENTRXFRAMEDROP signal, the nature of an incoming RX frame can be distinguished for system monitoring. See [Table 3-2](#).

**Table 3-2: EMACCLIENTRXDVLD and EMACCLIENTRXFRAMEDROP Values**

Result of an Incoming RX Frame	EMACCLIENTRXDVLD	EMACCLIENTRXFRAMEDROP
No frame received	0	0
Frame received and AF address match	1	0
Frame received and no AF address match	0	1
Frame received and no AF address match but AF disabled	1	1
Frame received and AF address match but AF disabled	1	0

The receiver statistics vector (bit 27) also indicates, for each RX frame, whether an AF match occurred or not. See [Receiver Statistics Vector](#) for more information.

### Address Filter Attributes

The unicast address register, pause frame source address, and address filter promiscuous mode bit are set by attributes when the FPGA is configured. In this way, the address filter

performs functions with the unicast address without using the host interface. The four general address register values are not included as attributes.

When the host interface is used, all the address filter registers are accessible by software, using either the DCR bus or the generic host bus. The attribute settings of the registers can be overridden by the software through the host interface. Also, the four general address registers are programmed through the host interface.

## Client RX Data/Control Interface

The AF generates the EMACCLIENTRXFRAMEDROP signal to inform the client that the destination MAC address of an incoming receive Ethernet frame does not match any of the acceptable addresses stored in the AF. This control signal is asserted regardless of whether the AF is enabled or disabled (promiscuous mode).

Figure 3-23 shows the timing diagram when a frame matches a valid location in the AF (8-bit mode).

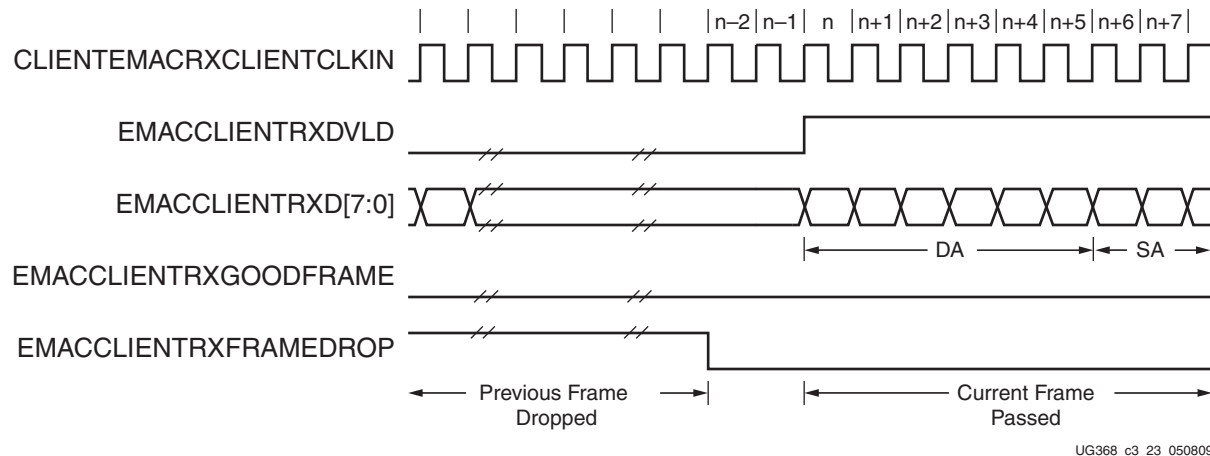


Figure 3-23: Frame Matching Timing Diagram (8-Bit Mode)

Figure 3-24 shows the timing diagram when a frame matches a valid location in the AF (16-bit mode).

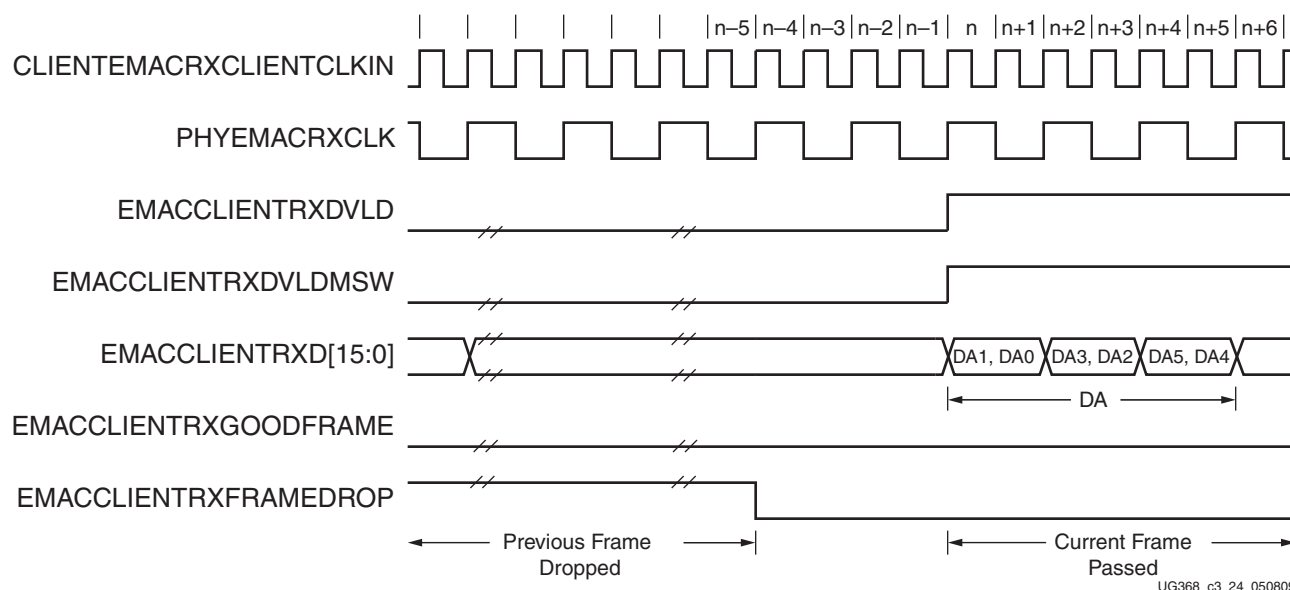


Figure 3-24: Frame Matching Timing Diagram (16-Bit Mode)

Figure 3-25 shows the timing diagram when a frame fails to match a valid location in the AF (8-bit mode) and the frame drop signal is generated.

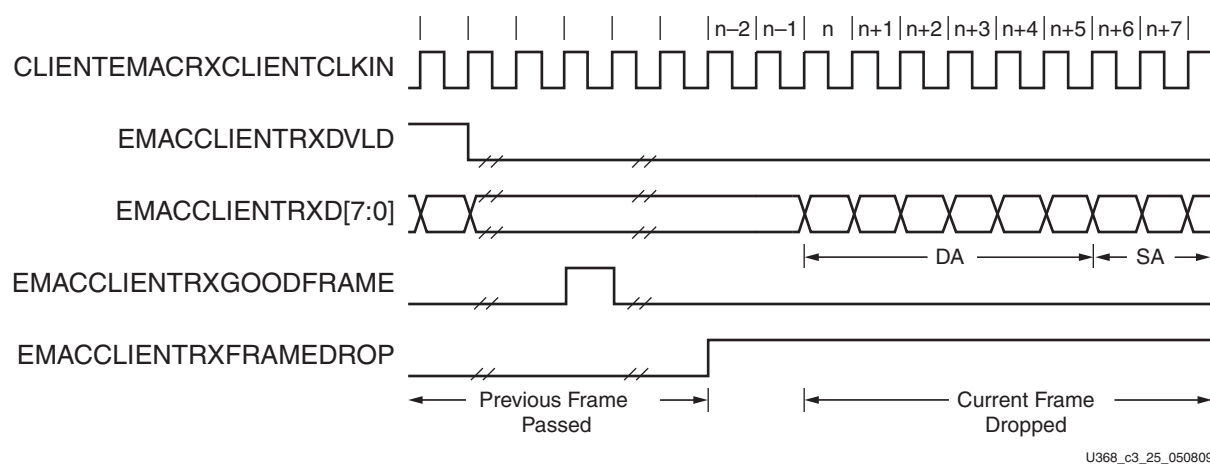


Figure 3-25: Frame Matching Failed Timing Diagram (8-Bit Mode)

Figure 3-26 shows the timing diagram when a frame fails to match a valid location in the AF (16-bit mode) and the frame drop signal is generated.

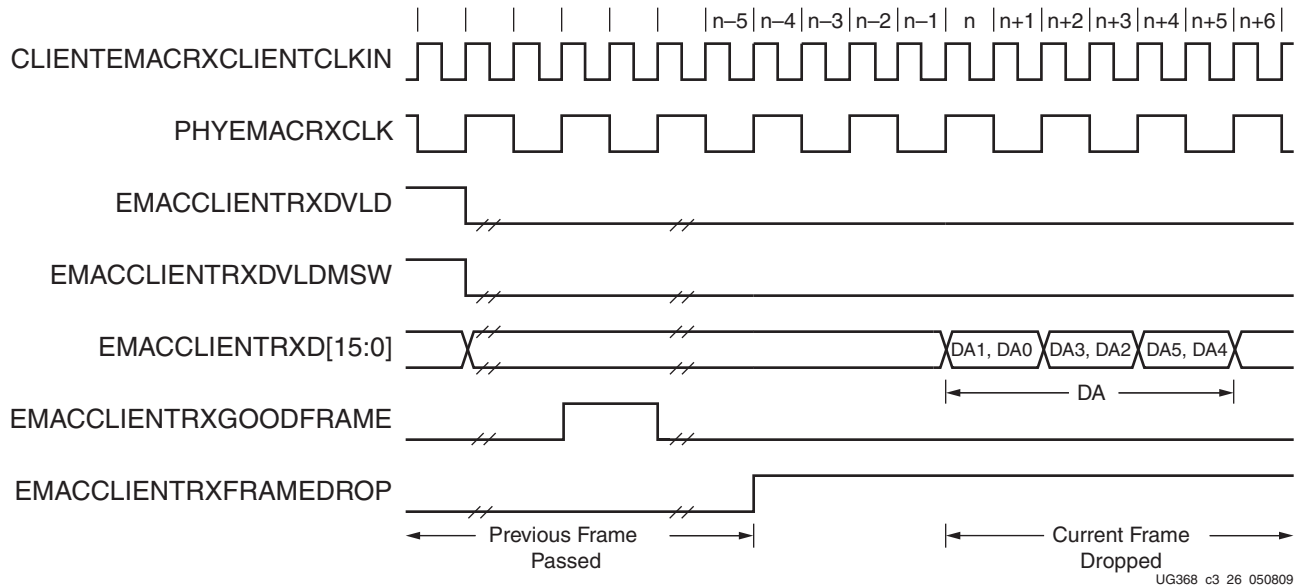


Figure 3-26: Frame Matching Failed Timing Diagram (16-Bit Mode)

## Flow Control Block

The flow control block is designed to clause 31 of the IEEE Std 802.3-2005 standard. The Ethernet MAC can be configured to send PAUSE frames and to act upon the PAUSE frame's reception during full-duplex operation. These two behaviors can be configured asymmetrically (see [Configuration Registers, page 84](#)).

### Requirement for Flow Control

[Figure 3-27](#) illustrates the requirement for flow control. The Ethernet MAC on the right side of the figure has a reference clock slightly faster than the nominal 125 MHz. The Ethernet MAC on the left side of the figure has a reference clock slightly slower than the nominal 125 MHz. This results in the Ethernet MAC on the left side of the figure not being able to match the full line rate of the Ethernet MAC on the right side (due to clock tolerances). The left Ethernet MAC is illustrated as performing a loopback implementation; this results in the FIFO filling up over time. Without flow control, this FIFO eventually fills and overflows, resulting in the corruption or loss of Ethernet frames.

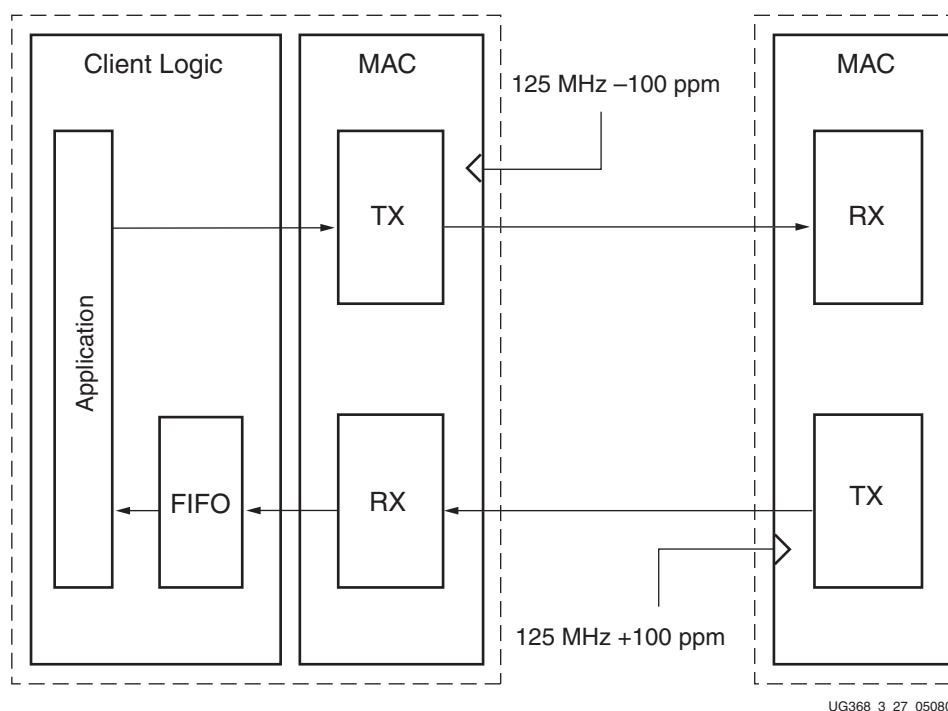


Figure 3-27: Requirement for Flow Control

## Flow Control Basics

An Ethernet MAC transmits a PAUSE control frame for the link partner to cease transmission for a defined period of time. For example, the left Ethernet MAC of Figure 3-27 initiates a pause request when the client FIFO (illustrated) reaches a nearly full state.

An Ethernet MAC responds to received PAUSE control frames by ceasing transmission of frames for the period of time defined in the received PAUSE control frame. For example, the right Ethernet MAC of Figure 3-27 ceases transmission after receiving the PAUSE control frame transmitted by the left Ethernet MAC. In a well-designed system, the right Ethernet MAC ceases transmission before the client FIFO of the left Ethernet MAC overflows. This provides time to empty the FIFO to a safe level before normal operation resumes. It also safeguards the system against FIFO overflow conditions and frame loss.

## Transmitting a PAUSE Control Frame

The client initiates a flow control frame by asserting CLIENTEMACPAUSEREQ when the pause value is on the CLIENTEMACPAUSEVAL[15:0] bus. When the optional clock enables are not used, these signals are synchronous to CLIENTEMACTXCLIENTCLKIN. The timing is shown in Figure 3-28.

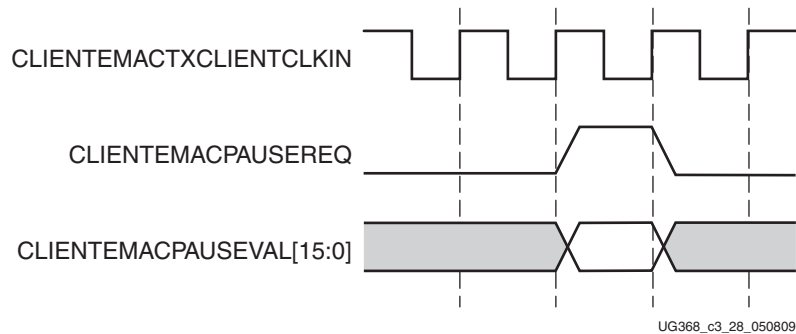


Figure 3-28: Pause Request Timing

When the optional clock enable clocking scheme is used, the signals are synchronous to PHYEMACTXGMIIMICLKIN. These signals are written into the Ethernet MAC when the enable signal, EMACCLIENTTXCLIENTCLKOUT, is High.

When the Ethernet MAC is configured to support transmit flow control, a PAUSE control frame is transmitted on the link. When CLIENTEMACPAUSEREQ is asserted, the PAUSE parameter is set to the CLIENTEMACPAUSEVAL[15:0] value. This does not disrupt any frame transmission in progress, but it takes priority over any pending frame transmission. The PAUSE control frame is transmitted even if the transmitter is in a paused state. An example of a PAUSE frame (not drawn to scale) is shown in Figure 3-29.

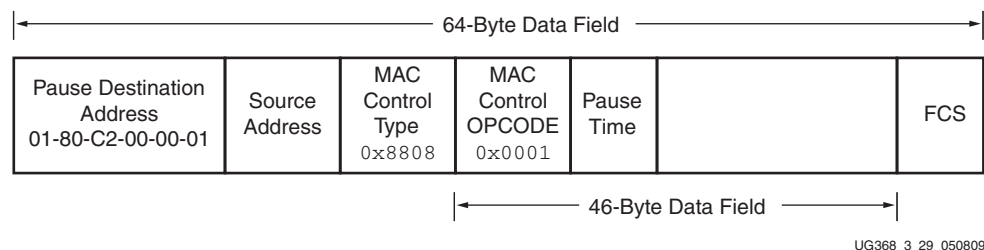


Figure 3-29: PAUSE Frame Example

The pause source address can be configured using the [Configuration Registers, page 84](#). The pause time transmitted in the PAUSE frame is the current sampled CLIENTEMACPAUSEVAL[15:0] value.

Because only a single PAUSE control frame request is stored by the transmitter, if CLIENTEMACPAUSEREQ is asserted multiple times prior to the transmission of a PAUSE control (i.e., while the previous frame transmission is still in progress), only a single PAUSE frame is transmitted.

## Receiving a PAUSE Control Frame

When an error-free frame is received by the Ethernet MAC, it examines the following information:

- The destination address field is matched against the Ethernet MAC control multicast address (01-80-C2-00-00-01) and the configured address for the Ethernet MAC (see [Address Filter Registers](#), page 89).
- The LT field is matched against the Ethernet MAC control type code.
- If the second match is TRUE, the OPCODE field contents are matched against the Ethernet MAC control OPCODE for pause frames.

If the frame passes all of the previous checks, is of minimum legal size, and the Ethernet MAC flow control logic for the receiver is enabled, then the pause value parameter in the frame is used to inhibit transmitter operation for a time defined in the IEEE Std 802.3-2005 specification. This inhibit is implemented by delaying the assertion of EMACCLIENTTXACK at the transmitter client interface until the requested pause duration has expired. Because the received PAUSE frame has been acted upon, it is passed to the client with EMACCLIENTRXBADFRAME asserted, indicating to the client that the frame should be dropped.

If the second match is TRUE and the frame is not exactly 64 bytes in length (when control frame length checking is enabled), the reception of any frame is considered to be an invalid control frame. This frame is ignored by the flow control logic and passed to the client with EMACCLIENTRXBADFRAME asserted. EMACCLIENTRXBADFRAME will be asserted even if flow control is not enabled.

If any of the previous matches are FALSE or if the Ethernet MAC flow control logic for the receiver is disabled, the frame is ignored by the flow control logic and is passed on to the client. It is the responsibility of the receiver client logic to drop any frames where the LT field matches the Ethernet MAC control type code. The receiver statistics vector (bit 19) indicates if the previous frame is a control frame. See [Receiver Statistics Vector](#) for more information.

## Flow Control Implementation Example

In the system illustrated in [Figure 3-27](#), the Ethernet MAC on the left side of the figure cannot match the full line rate of the right Ethernet MAC due to clock tolerances. Over time, the FIFO fills and overflows.

This example implements a flow control method to reduce, over a long time period, the full line rate of the right Ethernet MAC to less than the full line rate capability of the left Ethernet MAC.

### Method

1. Choose a FIFO with a nearly full occupancy threshold. A 7/8 occupancy is used in this description, but the choice of threshold is implementation specific. When the occupancy of the FIFO exceeds this occupancy, initiate a single PAUSE control frame with 0xFFFF used as the pause\_quantum duration (0xFFFF is placed on pause\_val[15:0]). This is the maximum pause duration and causes the right Ethernet MAC to cease transmission and the FIFO of the left Ethernet MAC to start emptying.
2. Choose a second FIFO with an occupancy threshold of 1/2 (the choice of threshold is implementation specific). When the occupancy of the FIFO falls below this occupancy, initiate a second PAUSE control frame with 0x0000 used as the pause\_quantum



duration (0x0000 is placed on pause\_val[15:0]). This indicates a zero pause duration, and upon receiving this PAUSE control frame, the right Ethernet MAC immediately resumes transmission (there is no wait for the original requested pause duration to expire). This PAUSE control frame can, therefore, be considered a PAUSE Cancel command.

## Operation

Figure 3-30 illustrates the FIFO occupancy over time.

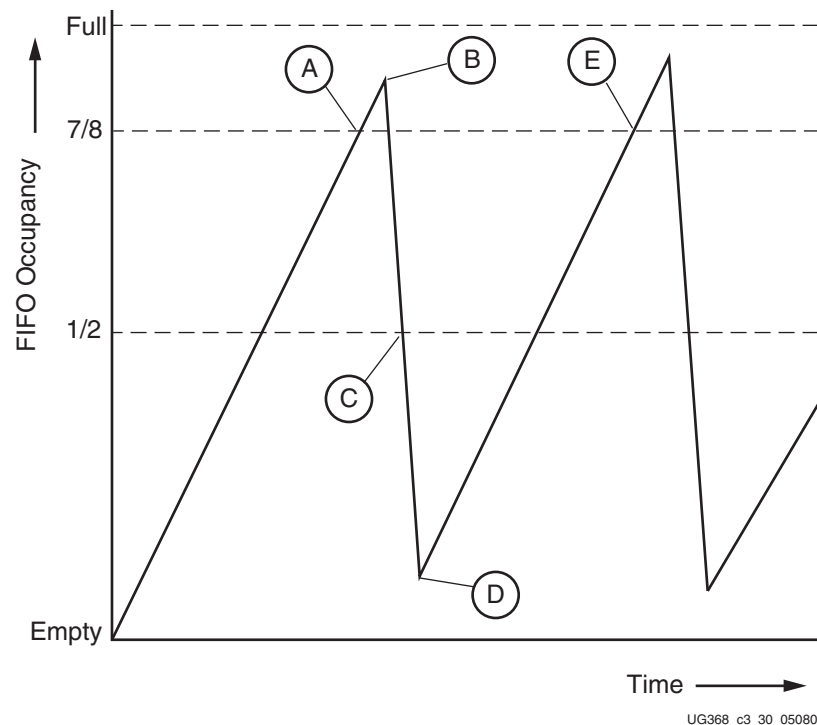


Figure 3-30: Flow Control Implementation Triggered from FIFO Occupancy

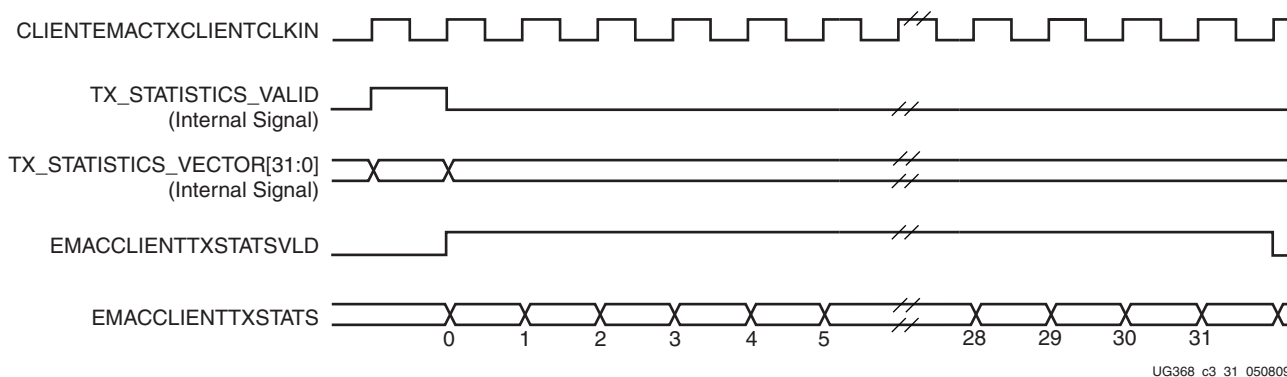
- The FIFO occupancy of the left Ethernet MAC gradually increases over time due to the clock tolerances. At point A, the occupancy has reached the threshold of 7/8 full, triggering the maximum duration PAUSE control frame request.
- Upon receiving the PAUSE control frame, the right Ethernet MAC ceases transmission at point B. The system designer should ensure sufficient space in the FIFO above the threshold (point A) to allow the transmitter to stop.
- After the right Ethernet MAC ceases transmission, the occupancy of the FIFO attached to the left Ethernet MAC empties. The occupancy falls to the second threshold of 1/2 occupancy at point C, triggering the zero duration PAUSE control frame request (the Pause Cancel command).
- Upon receiving this second PAUSE control frame, the right hand Ethernet MAC resumes transmission. To create an efficient system with the maximum possible line rate, the system designer should ensure sufficient space in the FIFO, below the threshold (point C), to prevent the FIFO emptying before transmission starts at point D.
- Normal operation resumes, and the FIFO occupancy again gradually increases over time. At point E, the flow control cycle repeats.

## Statistics Vectors

### Transmitter Statistics Vector

TX\_STATISTICS\_VECTOR contains the statistics for the frame transmitted. The TX\_STATISTICS\_VECTOR is a 32-bit vector and an internal signal. This vector is muxed out to a one-bit signal, EMACCLIENTTXSTATS, as shown in [Figure 3-31](#), following frame transmission. For standard clock management, the vector is driven synchronously by the transmitter clock, CLIENTEMACTXCLIENTCLKIN. For alternative clock management, including the use of clock enables, the vector is driven synchronously to other TX client logic signals. See [Chapter 6, Physical Interface](#), for relevant clocking networks.

The bit-field definition for the transmitter statistics vector is defined in [Table 3-3, page 76](#).



**Figure 3-31: Transmitter Statistics MUX Timing**

The block diagram for the transmitter statistics MUX in the Ethernet MAC is shown in [Figure 3-32](#).

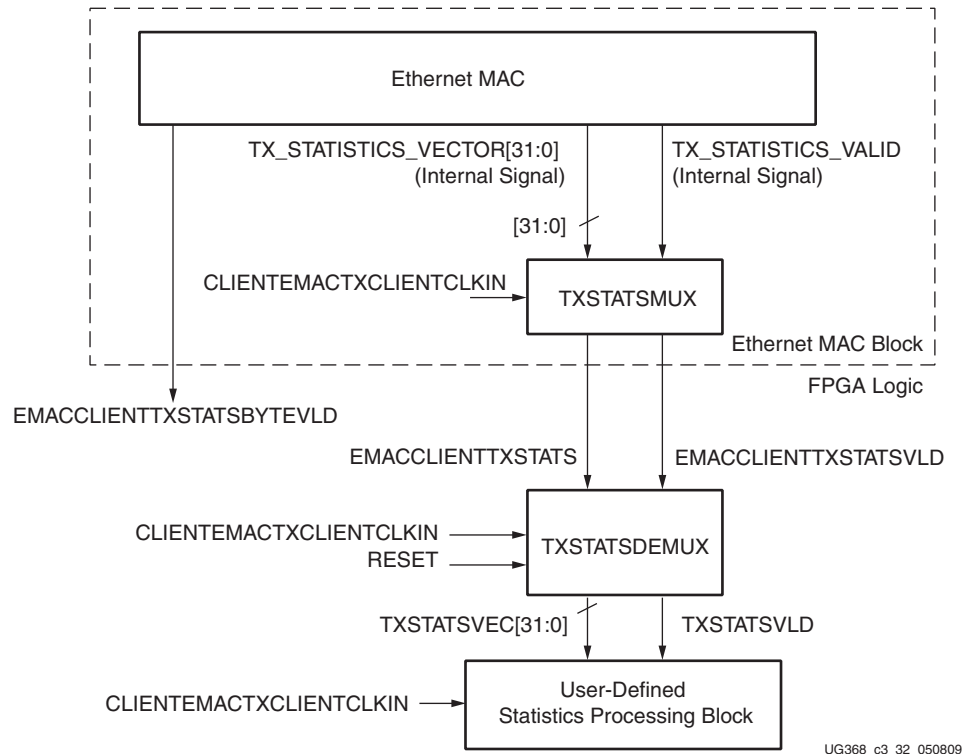


Figure 3-32: Transmitter Statistics MUX Block Diagram

All bit fields in EMACCLIENTTXSTATS are only valid when EMACCLIENTTXSTATSVLD is asserted as illustrated in Figure 3-33. EMACCLIENTTXSTATSBYTEVLD is asserted if an Ethernet MAC frame byte (DA to FCS inclusive) is being transmitted. The signal is valid on every CLIENTEMACTXCLIENTCLKIN cycle.

TX\_STATISTICS\_VECTOR (bits 28 down to 20 inclusive) are only for half-duplex mode. When operating in full-duplex mode, these bits are reset to a logic 0.

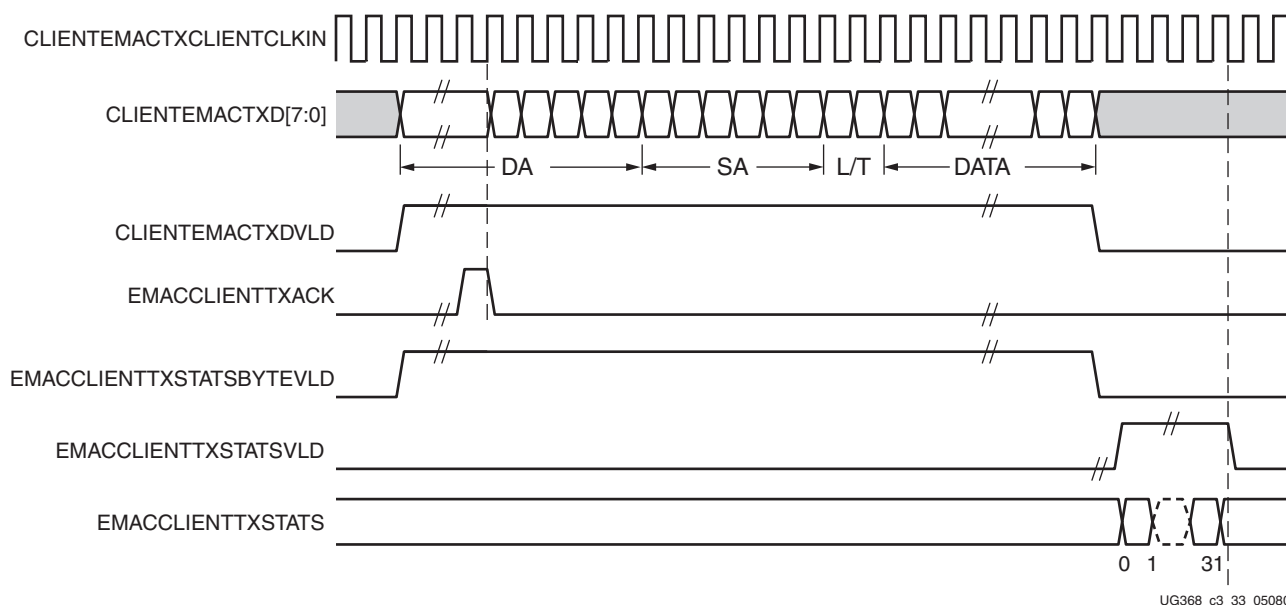


Figure 3-33: Transmitter Statistics Vector Timing

Table 3-3: Bit Definitions for the Transmitter Statistics Vector

TX_STATISTICS_VECTOR	Name	Description
31	PAUSE_FRAME_TRANSMITTED	Asserted if the previous frame was a pause frame initiated by the Ethernet MAC in response to asserting CLIENTEMACPAUSEREQ.
30	Reserved	Undefined.
29	Reserved	Returns a logic 0.
[28:25] <sup>(1)</sup>	TX_ATTEMPTS[3:0]	The number of attempts made to transmit the previous frame. A 4-bit number where 0 is one attempt; 1 is two attempts...up to 15 to describe 16 attempts.
24 <sup>(1)</sup>	Reserved	Returns a logic 0.
23 <sup>(1)</sup>	EXCESSIVE_COLLISION	Asserted if a collision is detected on each of the last 16 attempts to transmit the previous frame.
22 <sup>(1)</sup>	LATE_COLLISION	Asserted if a late collision occurred during frame transmission.
21 <sup>(1)</sup>	EXCESSIVE_DEFERRAL	Asserted if the previous frame was deferred for an excessive amount of time as defined by the maxDeferTime constant in the IEEE Std 802.3-2005 specification.
20 <sup>(1)</sup>	TX_DEFERRED	Asserted if transmission of the frame was deferred.
19	VLAN_FRAME	Asserted if the previous frame contains a VLAN identifier in the LT field when transmitter VLAN operation is enabled.

Table 3-3: Bit Definitions for the Transmitter Statistics Vector (Cont'd)

TX_STATISTICS_VECTOR	Name	Description
[18:5]	FRAME_LENGTH_COUNT	The length of the previous frame in number of bytes. The count sticks at 16383 for jumbo frames larger than this value. Stops at 16383.
4	CONTROL_FRAME	Asserted if the previous frame has the special Ethernet MAC control type code 88-08 in the LT field.
3	UNDERRUN_FRAME	Asserted if the previous frame contains an underrun error.
2	MULTICAST_FRAME	Asserted if the previous frame contains a multicast address in the destination address field.
1	BROADCAST_FRAME	Asserted if the previous frame contains a broadcast address in the destination address field.
0	SUCCESSFUL_FRAME	Asserted if the previous frame is transmitted without error.

**Notes:**

1. Bits 28:20 of TX\_STATISTICS\_VECTOR are valid for half-duplex mode only.

## Receiver Statistics Vector

RX\_STATISTICS\_VECTOR contains the statistics for the frame received. It is a 28-bit vector and an internal signal. This vector is muxed out to a 7-bit signal EMACCLIENTRXSTATS[6:0], as shown in Figure 3-34, following frame reception. For standard clock management, the vector is driven synchronously by the receiver clock, CLIENTEMACRXCLIENTCLKIN. For alternative clock management, including the use of clock enables, the vector is driven synchronously to other RX client logic signals. See Chapter 6, Physical Interface, for relevant clocking networks.

The bit-field definition for the receiver statistics vector is defined in Table 3-4.

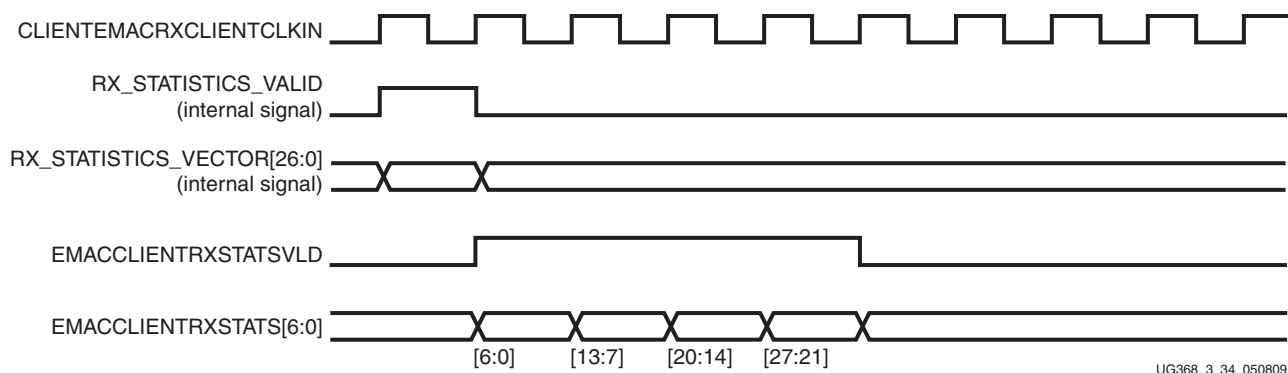


Figure 3-34: Receiver Statistics MUX Timing

The block diagram for the receiver statistics MUX in the Ethernet MAC is shown in Figure 3-35.

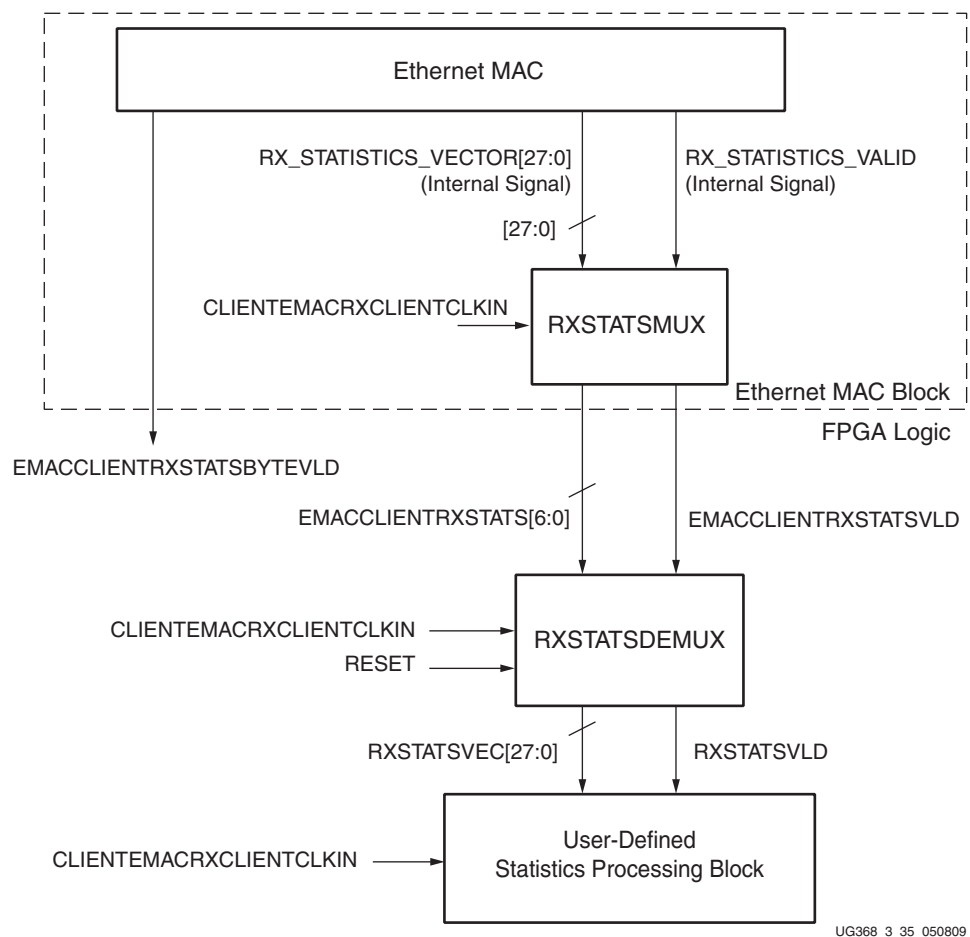
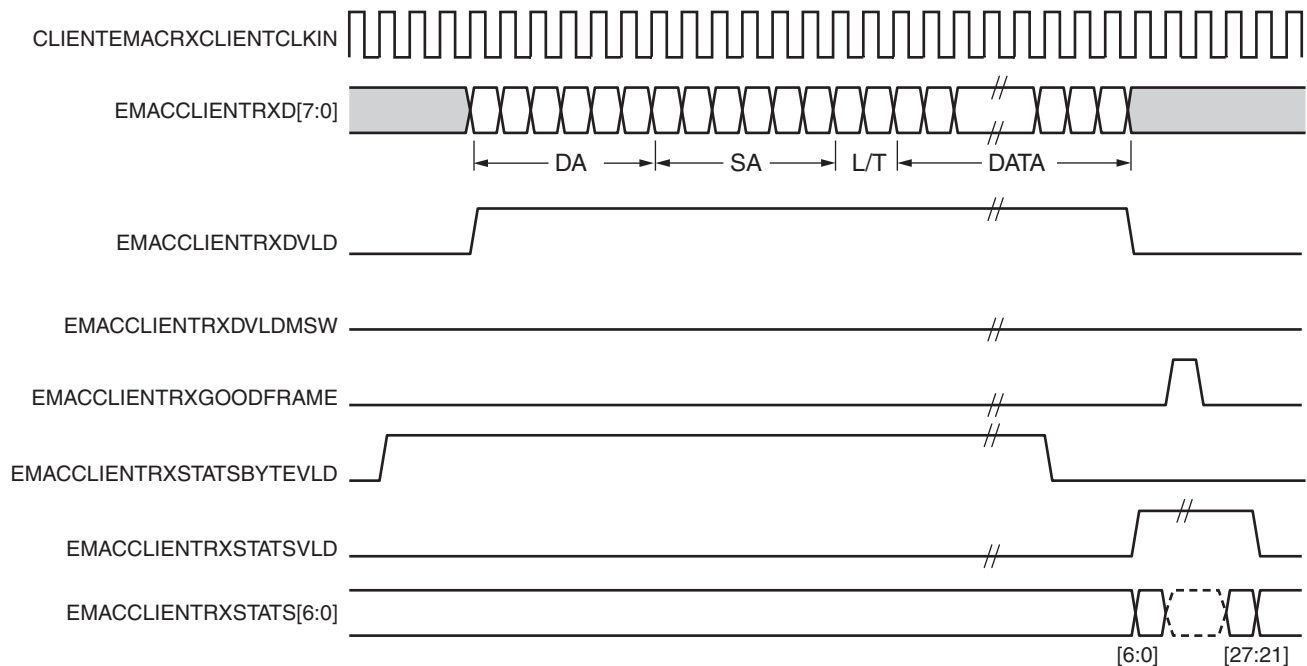


Figure 3-35: Receiver Statistics MUX Block Diagram

All bit-fields for `EMACCLIENTRXSTATS[6:0]` are valid only when `EMACCLIENTRXSTATSVLD` is asserted as illustrated in Figure 3-36. `EMACCLIENTRXSTATSBYTEVLD` is asserted if an Ethernet MAC frame byte (DA to FCS inclusive) is received. This is valid on every `CLIENTEMACRXCLIENTCLKIN` cycle.



UG368\_c3\_36\_050809

Figure 3-36: Receiver Statistics Vector Timing

Table 3-4: Bit Definitions for the Receiver Statistics Vector

RX_STATISTICS_VECTOR	Name	Description
27	ADDRESS_MATCH	When the Ethernet MAC is configured in Address Filtering mode, asserted if the previous frame successfully passed the address filter. When the Ethernet MAC is configured in promiscuous mode, this bit is always asserted.
26	ALIGNMENT_ERROR	Used in 10/100 MII mode. Asserted if the previous frame received has an incorrect FCS value and a misalignment occurs when the 4-bit MII data bus is converted to the 8-bit GMII data bus.
25	Length/Type Out of Range	Asserted if the LT field contains a length value that does not match the number of Ethernet MAC client data bytes received. Also asserted High if the LT field indicates that the frame contains padding but the number of Ethernet MAC client data bytes received is not equal to 64 bytes (minimum frame size).  This bit is not defined when LT field error-checks are disabled or when received frames are less than the legal minimum length.
24	BAD_OPCODE	Asserted if the previous frame is error free. Contains the special control frame identifier in the LT field, but contains an OPCODE unsupported by the Ethernet MAC (any OPCODE other than PAUSE).

Table 3-4: Bit Definitions for the Receiver Statistics Vector (Cont'd)

RX_STATISTICS_VECTOR	Name	Description
23	FLOW_CONTROL_FRAME	Asserted if the previous frame is error-free. Contains the special control frame identifier in the LT field. Contains a destination address matching either the Ethernet MAC control multicast address or the configured source address of the Ethernet MAC. Contains the supported PAUSE OPCODE and is acted upon by the Ethernet MAC.
22	Reserved.	Undefined.
21	VLAN_FRAME	Asserted if the previous frame contains a VLAN identifier in the LT field when receiver VLAN operation is enabled.
20	OUT_OF_BOUNDS	Asserted if the previous frame exceeded the specified IEEE Std 802.3-2005 maximum legal length (see <a href="#">Maximum Permitted Frame Length/Jumbo Frames for Reception</a> , page 61). This is only valid if jumbo frames are disabled.
19	CONTROL_FRAME	Asserted if the previous frame contains the special control frame identifier in the LT field.
[18:5]	FRAME_LENGTH_COUNT	The length of the previous frame in number of bytes. The count sticks at 16383 for any jumbo frames larger than this value.
4	MULTICAST_FRAME	Asserted if the previous frame contains a multicast address in the destination address field.
3	BROADCAST_FRAME	Asserted if the previous frame contained the broadcast address in the destination address field.
2	FCS_ERROR	Asserted if the previous frame received has an incorrect FCS value or the Ethernet MAC detects error codes during frame reception.
1	BAD_FRAME <sup>(1)</sup>	Asserted if the previous frame received contains errors.
0	GOOD_FRAME <sup>(1)</sup>	Asserted if the previous frame received is error-free.

**Notes:**

1. If the length/type field error checks are disabled, then a frame containing this type of error is marked as a GOOD\_FRAME, providing no additional errors were detected.

## Statistics Registers/Counters

The Ethernet MAC does not collect statistics on the success and failure of various operations. A custom statistics counter can be implemented in the FPGA logic to collect the statistics. A parameterizable Ethernet statistics core is available from the LogiCORE™ libraries. For more information, see:

[http://www.xilinx.com/products/ipcenter/ETHERNET\\_STATS.htm](http://www.xilinx.com/products/ipcenter/ETHERNET_STATS.htm).

When the DCR is used, it can access the statistics counter registers in the FPGA logic through the DCR bridge in the host interface. [Chapter 7, Interfacing to a Statistics Block](#), describes how to integrate Ethernet Statistics LogiCORE core into a design, as well as how to access the statistics registers using either the host bus or the DCR bus.



# *Host/DCR Bus Interfaces*

---

This chapter contains the following sections:

- [Introduction to the Ethernet MAC Host Interface](#)
- [Ethernet MAC Register Descriptions](#)
- [Using the Host Bus](#)
- [Using the DCR Bus](#)

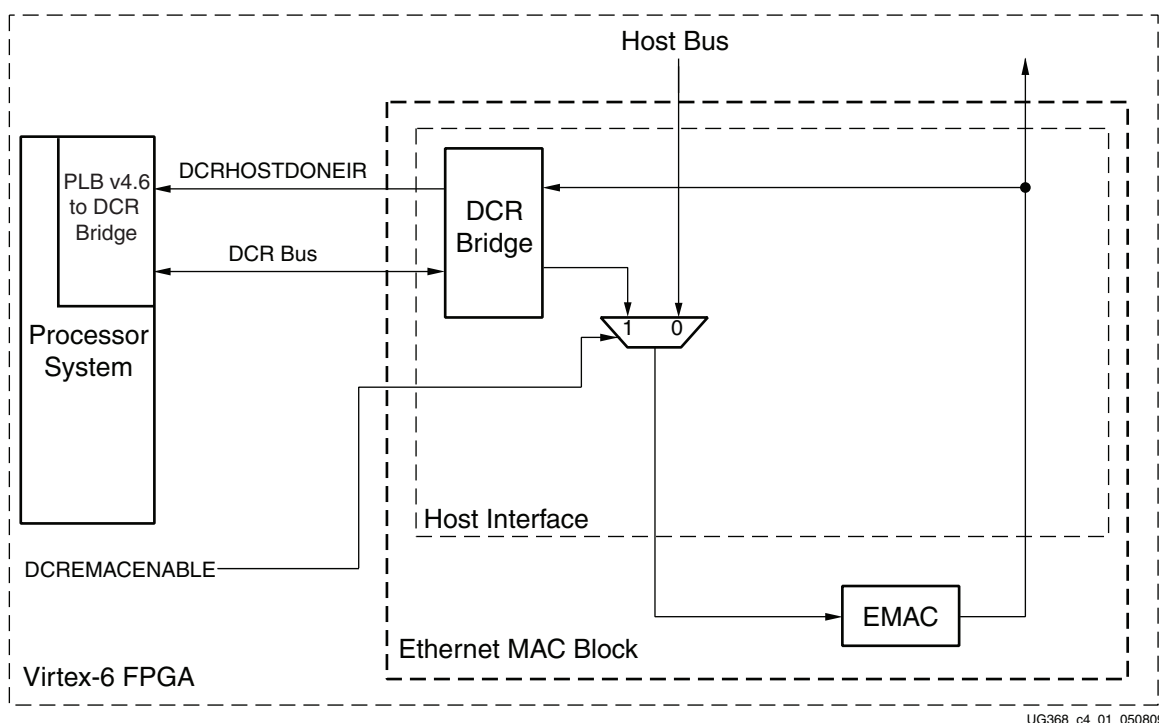
## **Introduction to the Ethernet MAC Host Interface**

Access to the Ethernet MAC through the host interface allows the user to:

- Read and write Ethernet MAC configuration registers
- Read and write Ethernet MAC address filter registers
- Read and write the PCS/PMA sublayer registers
- Drive the Management Data I/O (MDIO) interface to control an external device

The Ethernet MAC host interface can be accessed using either the host bus or the DCR bus. The Ethernet MAC should be configured to use the required bus by setting the DCREMACENABLE input bit accordingly, see [DCR Bus Signals](#), page 32.

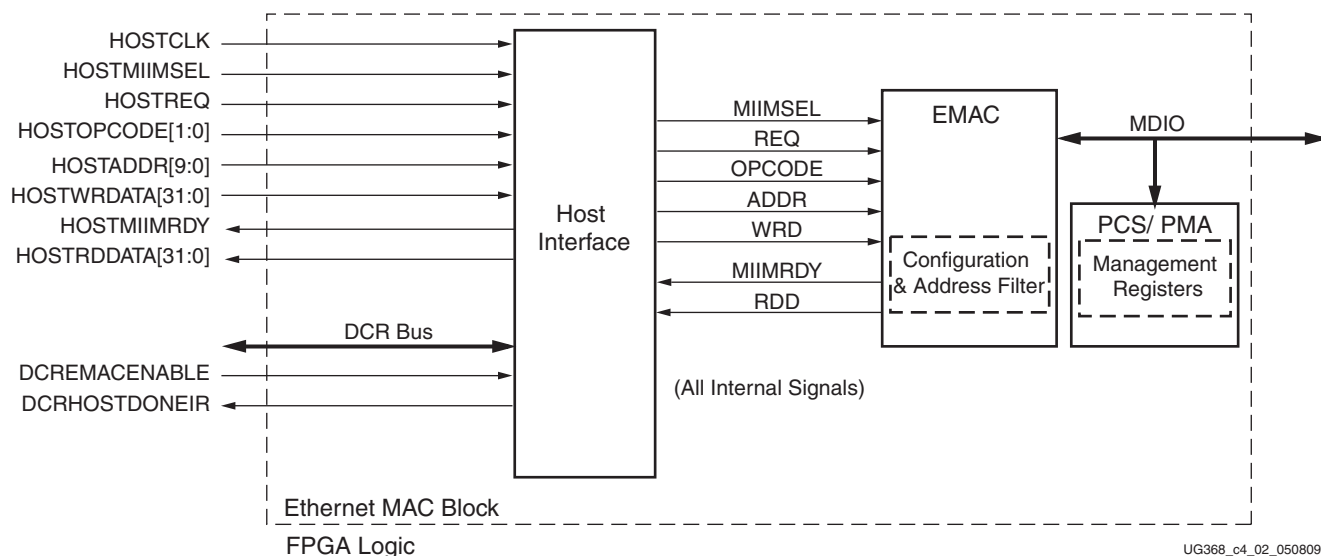
Figure 4-1 shows the internal structure of the host interface.



**Figure 4-1: Host Interface**

The XPS\_LL\_TEMAC soft core delivered through the XPS tool controls the Ethernet MAC via the processor local bus (PLB) v4.6 interface. A bridge is provided to convert between the PLB v4.6 and DCR bus standards. For more information on the PLB v4.6 interface, see [DS531](#), *Processor Local Bus (PLB) v4.6 (v1.00a)*.

Figure 4-2 shows the block diagram of the host interface.



**Figure 4-2: Ethernet MAC Host Interface Block Diagram**

To access management registers in the PCS/PMA sublayer, the MDIO interface is used. The MDIO interface is described in detail in [Chapter 5, MDIO Interface](#), and access to the MDIO through the host interface is discussed later in this chapter.

## Ethernet MAC Register Descriptions

Each Ethernet MAC has twelve configuration and address filter registers. These registers are accessed through the host interface and can be written to at any time. Both the receiver and transmitter logic only responds to configuration changes during interframe gaps. The configurable resets are the only exception that take effect immediately. Writes made to the address filter registers during the interframe gap take effect immediately, whereas writes made during frame reception will affect only subsequent frames.

**Table 4-1: Ethernet MAC Register Addresses**

ADDRESS[9:0]	Register Description
0x200	Receiver Configuration (Word 0)
0x240	Receiver Configuration (Word 1)
0x280	Transmitter Configuration
0x2C0	Flow Control Configuration
0x300	Ethernet MAC Mode Configuration
0x320	RGMII/SGMII Configuration
0x340	Management Configuration
0x380	Unicast Address (Word 0)
0x384	Unicast Address (Word 1)
0x388	Additional Address Table Access (Word 0)
0x38C	Additional Address Table Access (Word 1)
0x390	Address Filter Mode

## Configuration Registers

The Ethernet MAC configuration registers and the contents of the registers are shown in Table 4-2 through Table 4-8.

**Table 4-2: Receiver Configuration Register (Word 0)**

0x200	<div style="display: flex; justify-content: space-between;"> <span>MSB</span> <span>LSB</span> </div> <div style="display: flex; justify-content: space-between;"> <span>31</span><span>30</span><span>29</span><span>28</span><span>27</span><span>26</span><span>25</span><span>24</span><span>23</span><span>22</span><span>21</span><span>20</span><span>19</span><span>18</span><span>17</span><span>16</span><span>15</span><span>14</span><span>13</span><span>12</span><span>11</span><span>10</span><span>9</span><span>8</span><span>7</span><span>6</span><span>5</span><span>4</span><span>3</span><span>2</span><span>1</span><span>0</span> </div>																															
	PAUSE_FRAME_ADDRESS[31:0]																															

Bit	Description	Default Value	R/W
[31:0]	<p>PAUSE Frame Ethernet MAC Address [31:0]. This address is used to match the Ethernet MAC against the destination address of any incoming flow control frames. It is also used by the flow control block as the source address for any outbound flow control frames.</p> <p>The address is ordered so the first byte transmitted/received is the lowest positioned byte in the register; for example, a MAC address of AA-BB-CC-DD-EE-FF is stored in address [47:0] as 0xFFEEDDCCBBAA.</p> <p>Tie to the same value as EMAC_UNICASTADDR[31:0].</p>	EMAC_PAUSEADDR[31:0]	R/W

**Table 4-3: Receiver Configuration Register (Word 1)**

0x240

MSB

LSB

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RST	JUM	FCS	RX	VLAN	HD	LT_DIS	CL_DIS	RESERVED								PAUSE_FRAME_ADDRESS[47:32]															

Bit	Description	Default Value	R/W
[15:0]	Pause frame Ethernet MAC Address [47:32]. Tie to the same value as EMAC_UNICASTADDR[47:32].	EMAC_PAUSEADDR[47:32]	R/W
[23:16]	Reserved.	—	
[24]	Control frame length check disable. When this bit is 1, control frames larger than the minimum frame length can be accepted.	EMAC_CTRLLENCHECK_DISABLE	R/W
[25]	Length/Type Check disable. When this bit is 1, it disables the Length/Type field check on the frame.	EMAC_LTCHECK_DISABLE	R/W
[26]	Half-duplex enable (applicable in 10/100 Mb/s mode only, for certain physical interfaces). When this bit is 1, the receiver operates in half-duplex mode. When the bit is 0, the receiver operates in full-duplex mode.	EMAC_RXHALFDUPLEX	R/W
[27]	VLAN enable. When this bit is 1, the receiver accepts VLAN tagged frames. The maximum payload length increases by four bytes.	EMAC_RXVLAN_ENABLE	R/W

**Table 4-4: Transmitter Configuration Register**

Bit	Description	Default Value	R/W
[24:0]	Reserved.	–	
[25]	IFG adjustment enable. When this bit is 1, the transmitter reads the value of CLIENTEMACTXIFGDELAY at the start of frame transmission and adjusts the IFG, if operating in full-duplex mode.	EMAC_TXIFGADJUST_ENABLE	R/W
[26]	Half-duplex enable (applicable in 10/100 Mb/s mode only), for certain physical interfaces. When this bit is 1, the transmitter operates in half-duplex mode. When this bit is 0, the transmitter operates in full-duplex mode.	EMAC_TXHALFDUPLEX	R/W
[27]	VLAN enable. When this bit is 1, the transmitter allows transmission of the VLAN tagged frames.	EMAC_TXVLAN_ENABLE	R/W
[28]	Transmit enable. When this bit is 1, the transmitter is enabled for operation.	EMAC_TX_ENABLE	R/W
[29]	In-band FCS enable. When this bit is 1, the Ethernet MAC transmitter uses the FCS field supplied by the client.	EMAC_TXINBANDFCS_ENABLE	R/W
[30]	Jumbo frame enable. When this bit is 1, the transmitter can send frames greater than the maximum length specified in IEEE Std 802.3-2005. When this bit is 0, it only sends frames less than the specified maximum length.	EMAC_TXJUMBOFRAME_ENABLE	R/W
[31]	Reset. When this bit is 1, the transmitter is reset. The bit automatically reverts to 0. This reset also sets all of the transmitter configuration registers to their default values.	EMAC_TXRESET	R/W

Table 4-5: Flow Control Configuration Register

		MSB																												LSB			
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x2C0	RESERVED				RESERVED																												
	FCTX																																
	FCRX																																

Bit	Description	Default Value	R/W
[28:0]	Reserved.	—	
[29]	Flow control enable (RX). When this bit is 1, received flow control frames inhibit transmitter operation. When this bit is 0, flow control frames are passed to the client.	EMAC_RXFLOWCTRL_ENABLE	R/W
[30]	Flow control enable (TX). When this bit is 1, assertion of the CLIENTEMACPAUSEREQ signal causes a flow control frame to be sent from the transmitter. When this bit is 0, the CLIENTEMACPAUSEREQ signal has no effect.	EMAC_TXFLOWCTRL_ENABLE	R/W
[31]	Reserved.	—	

Table 4-6: Ethernet MAC Mode Configuration Register

0x300	MSB																														LSB		
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	LINK SPEED		RGMII	SGMII	GPCS	HOST	TX16	RX16	RESERVED																LINK TIMER								

Bit	Description	Default Value	R/W
[8:0]	Sets the programmable link timer value, for operation with 1000BASE-X or SGMII modes (see <a href="#">1000BASE-X Auto-Negotiation Link Timer</a> , page 166).	EMAC_LINKTIMERVAL[8:0]	R/W
[23:9]	Reserved.	—	
[24]	Receive 16-bit Client Interface enable. When this bit is 1, the receive data client interface is 16 bits wide. When this bit is 0, the receive data client interface is 8 bits wide. This bit is valid only when using 1000BASE-X PCS/PMA mode.	EMAC_RX16BITCLIENT_ENABLE	R
[25]	Transmit 16-bit Client Interface enable. When this bit is 1, the transmit data client interface is 16 bits wide. When this bit is 0, the transmit data client interface is 8 bits wide. This bit is valid only when using 1000BASE-X PCS/PMA mode.	EMAC_TX16BITCLIENT_ENABLE	R
[26]	Host Interface enable. When this bit is 1, the host interface is enabled. When this bit is 0, the host interface is disabled. <a href="#">Ethernet MAC Attributes</a> , page 36	EMAC_HOST_ENABLE	R

**Table 4-7: RGMII/SGMII Configuration Register**

Bit	Description	Default Value	R/W
[0]	RGMII link. Valid in RGMII mode configuration only. When this bit is 1, the link is up. When this bit is 0, the link is down. This displays the link information from the PHY to the Ethernet MAC, encoded by GMII_RX_DV and GMII_RX_ER during the IFG.	0	R
[1]	RGMII duplex status. Valid in RGMII mode configuration only. This bit is 0 for half-duplex and 1 for full-duplex. This displays the duplex information from the PHY to the Ethernet MAC, encoded by GMII_RX_DV and GMII_RX_ER during the IFG.	0	R
[3:2]	RGMII speed. Valid in RGMII mode configuration only. Link information from the PHY to the Ethernet MAC as encoded by GMII_RX_DV and GMII_RX_ER during the IFG. This two-bit vector is defined with the following values: <ul style="list-style-type: none"> <li>10 = 1000 Mb/s</li> <li>01 = 100 Mb/s</li> <li>00 = 10 Mb/s</li> <li>11 = N/A</li> </ul>	All 0s	R

Bit	Description	Default Value	R/W
[29:4]	Reserved	–	
[31:30]	SGMII speed. Valid in SGMII mode configuration only. This displays the SGMII speed information, as received by TX_CONFIG_REG[11:10] in the PCS/PMA register. See <a href="#">Table 5-19, page 128</a> . This two-bit vector is defined with the following values: <ul style="list-style-type: none"> <li>10 = 1000 Mb/s</li> <li>01 = 100 Mb/s</li> <li>00 = 10 Mb/s</li> <li>11 = N/A</li> </ul>	All 0s	R

Table 4-8: Management Configuration Register

0x34 0	<div>MSB<div>313029282726252423222120191817161514131211109876543210</div>LSB</div>																														
	RESERVED																								MDIOEN	CLOCK_DIVIDE[5:0]					

Bit	Description	Default Value	R/W
[5:0]	Clock divide [5:0]. This value is used to derive the EMACPHYMCLKOUT for external devices. See <a href="#">Chapter 5, MDIO Interface</a> .	All 0s	R/W
[6]	MDIO enable. When this bit is 1, the MDIO interface is used to access the PHY. This bit can only be asserted when clock divide is also loaded with a non-zero value. When this bit is 0, the MDIO interface is disabled, and the MDIO signals remain inactive. See <a href="#">Chapter 5, MDIO Interface</a> .	0	R/W
[31:7]	Reserved.	–	



Address Filter Registers

The Ethernet MAC Address filter has five address filter registers accessible through the host interface. One unicast address can be written into the unicast address register and four additional addresses can be written into the general address table, which is accessed indirectly through the General Address access registers. Figure 4-3 describes the General Address table access.

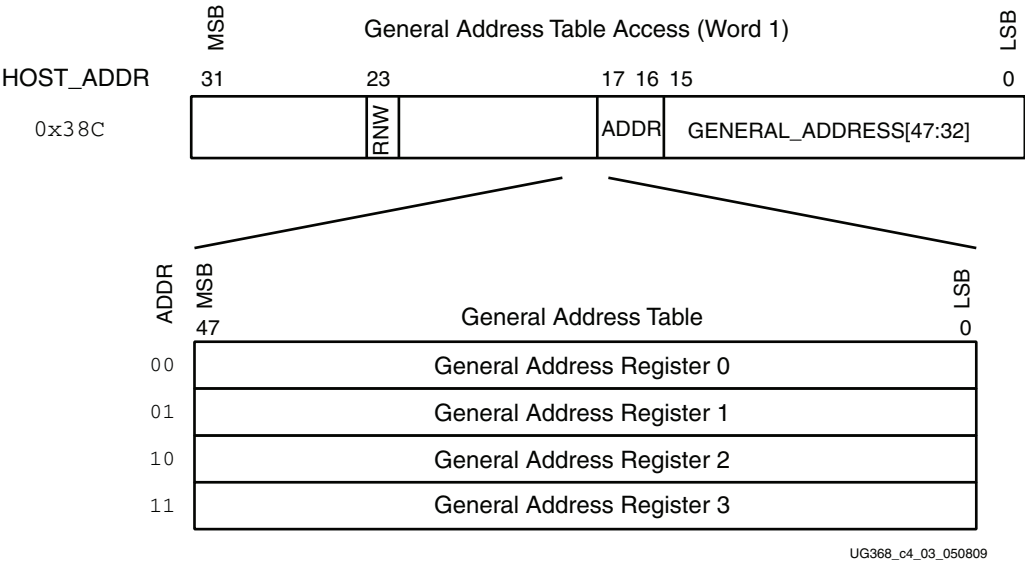


Figure 4-3: General Address Table Memory Diagram

Table 4-9: Unicast Address (Word 0)

0x380	<div> <div>MSB</div> <div>31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</div> <div>LSB</div> </div>	
	UNICAST_ADDRESS[31:0]	

Bit	Description	Default Value	R/W
[31:0]	<p>Unicast Address [31:0]. This address is used to match the Ethernet MAC against the destination address of any incoming frames.</p> <p>The address is ordered so the first byte transmitted/received is the lowest positioned byte in the register; for example, a MAC address of AA-BB-CC-DD-EE-FF is stored in address [47:0] as 0xFFEEDDCCBBAA.</p>	EMAC_UNICASTADDR[31:0]	R/W

Table 4-10: Unicast Address (Word 1)

0x384	<div>MSB<div>313029282726252423222120191817161514131211109876543210</div>LSB</div>																															
	RESERVED																UNICAST_ADDRESS[47:32]															

Bit	Description	Default Value	R/W
[15:0]	Unicast Address [47:32]	EMAC_UNICASTADDR[47:32]	R/W
[31:16]	Reserved.	—	

Table 4-11: Address Table Access (Word 0)

0x388	MSB	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB
	GENERAL_ADDRESS[31:0]																																	

Bit	Description	Default Value	R/W
[31:0]	General Address [31:0]. The general address bits [31:0] are temporarily deposited into this register for writing into a general address register. When a read from an address register has been performed, the address bits [31:0] are accessible from this register. The address is ordered so the first byte transmitted/received is the lowest positioned byte in the register; for example, a MAC address of AA-BB-CC-DD-EE-FF is stored in address [47:0] as 0xFFEEDDCCBBAA.	All 0s	R/W

Table 4-12: Address Table Access (Word 1)

0x38C	MSB																															LSB
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESERVED								RNW	RESERVED				ADDR	GENERAL_ADDRESS[47:32]																	

Bit	Description	Default Value	R/W
<b>General Address (Word 1)</b>			
[15:0]	General Address [47:32]. The general address bits [47:32] are temporarily deposited into this register for writing into a general address register.	All 0s	R/W
[17:16]	Address Table address. This two-bit vector is used to choose the general address register to access. <ul style="list-style-type: none"> <li>00 = General Address Register 0</li> <li>01 = General Address Register 1</li> <li>10 = General Address Register 2</li> <li>11 = General Address Register 3</li> </ul>	All 0s	R/W
[22:18]	Reserved.	—	
[23]	Address Table read enable (RNW). When this bit is 1, a general address register is read. When this bit is 0, a general address register is written with the address set in the general address table register.	0	R/W
[31:24]	Reserved.	—	

Table 4-13: Address Filter Mode

0x390

MSB

LSB

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PM	RESERVED																														

Bit	Description	Default Value	R/W
Address Filter Mode			
[30:0]	Reserved.	—	
[31]	Promiscuous Mode enable. When this bit is 1, the Address Filter block is disabled. When this bit is 0, the Address Filter block is enabled.	1: When EMAC_ADDRFILTERENABLE is FALSE 0: When EMAC_ADDRFILTERENABLE is TRUE	R/W

## Using the Host Bus

To access the host interface using the host bus, the DCREMACENABLE signal should be tied to logic 0.

When using the host bus, the type of transaction initiated is dependant on the host bus signals HOSTMIIMSEL and HOSTADDR[9]. [Table 4-14](#) shows the access method required for each transaction type.

Table 4-14: Host Bus Transaction Types

Transaction	R/W	HOSTMIIMSEL	HOSTADDR[9]	HOSTOPCODE[1]	HOSTOPCODE[0]
Configuration/ Address Filter	Read	0	1	1	X
	Write	0	1	0	X
MDIO access	Read	1	X	1	0
	Write	1	X	0	1

## Host Bus Clocking Requirements

The HOSTCLK can operate at speeds of up to 125 MHz. This clock can be tied to a user-supplied, 125 MHz input clock to save on clock resources, such as the 125 MHz clock provided into the PHYEMACGTCLK port (see [Appendix B, Ethernet MAC Clocks](#)).

The host clock (HOSTCLK) is used to derive the MDIO clock, MDC. Configuring the Ethernet MAC to derive MDC is detailed in [Accessing MDIO through the Host Interface, page 115](#).

## Reading and Writing MAC Configuration Registers

[Figure 4-4](#) shows the write timing for the configuration registers using the host bus. When accessing the configuration registers (i.e., when HOSTADDR[9] = 1 and HOSTMIIMSEL = 0), HOSTOPCODE[1] functions as an active-Low write enable signal, and HOSTOPCODE[0] is a don't care.

HOSTADDR[9:0] takes the address of the configuration register to be accessed, as described in Table 4-1.

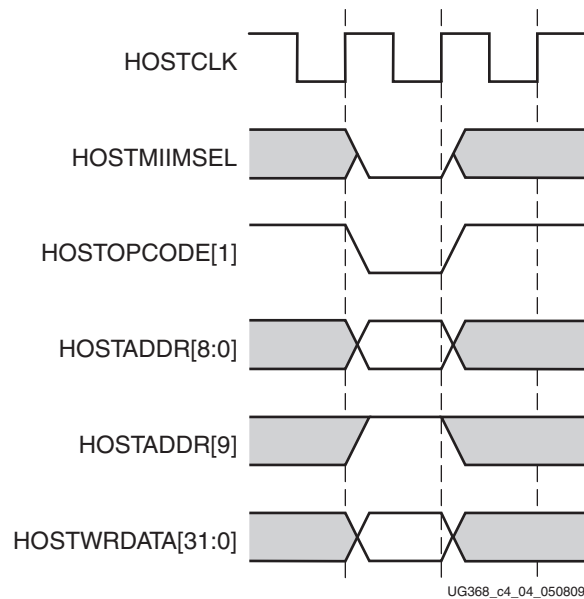


Figure 4-4: Configuration Register Write Timing

Figure 4-5 shows the read timing from the configuration registers. A High on HOSTOPCODE[1] indicates a read transaction on the bus. The contents of the register appear on HOSTRDDATA[31:0] on the HOSTCLK edge after the register address is asserted onto HOSTADDR. A Low on HOSTMIIMSEL indicates a configuration access. It must be held Low for an even number of clock cycles during a read operation.

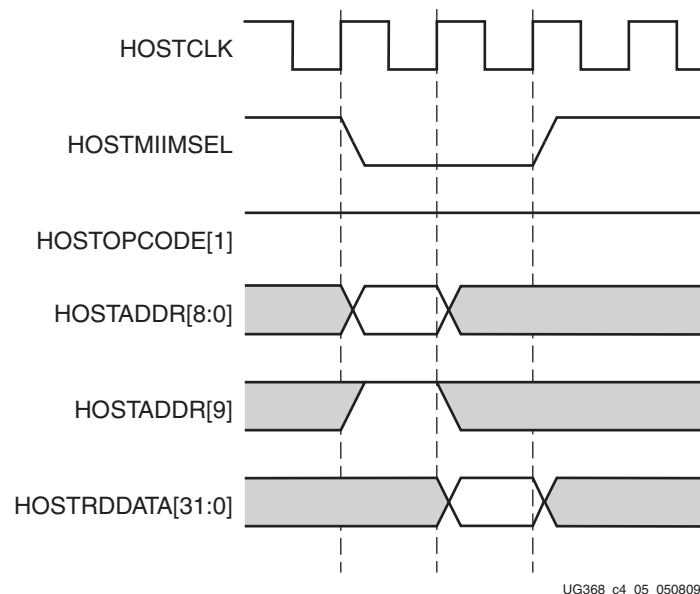


Figure 4-5: Configuration Register Read Timing

## Reading and Writing Address Filter Registers

Writing to registers in the Address Filter is the same as writing to configuration registers. The timing diagram shown for writing to the Ethernet MAC Configuration Registers (Figure 4-4) holds for Address Filter registers. HOSTADDR[9:0] takes the address of the address filter register to be accessed, as described in Table 4-1.

To write a general address register, two write operations are required. First bits [31:0] of the general address must be written into the General Address Table Access Word 0 register. The upper bits [47:32] of the general address must then be written into the General Address Table Access Word 1 register, along with the Address Table address and the RNW bit at logic 0.

Reading the Unicast Address registers and the Address Mode register is also the same as reading from configuration registers. The timing diagram shown for reading the Ethernet MAC configuration registers (Figure 4-5) applies to accesses for these registers.

Figure 4-6 shows the timing diagram for reading the general address from one of the four general address registers in the Address Table.

To read a general address register, a write must first be made to the General Address Table Access Word 1 register. The RNW field is set to 1, and the Address Table address field should be set with the register number to be read. The general address register data is returned in HOSTRDDATA[31:0] as shown in Figure 4-6. The LSW is the general address [31:0]. The MSW contains 0x0000 and the general address [47:32].

A Low on HOSTMIIMSEL indicates a configuration access. It must be held Low for an even number of clock cycles during a read operation.

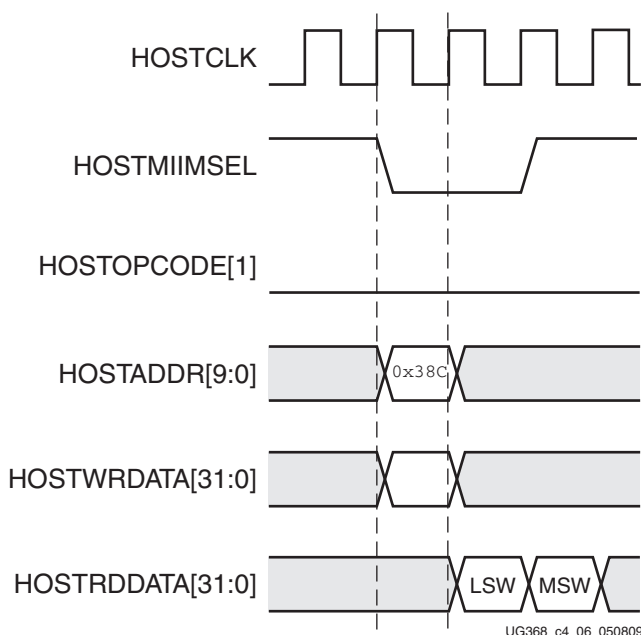


Figure 4-6: Address Filter General Address Register Read

## PCS/PMA Sublayer or External Device Access via MDIO

Access to the MDIO interface through the management interface is shown in the Figure 4-7 timing diagram. See Chapter 5, MDIO Interface, for a full description of the MDIO interface and PCS/PMA sublayer management registers. The MDIO interface must be

enabled using the Management Configuration Register (see Table 4-8). If the MDIO interface is not enabled, accesses complete in a single cycle (MIIMRDY remains High) but are ignored.

In MDIO transactions, the following applies:

- HOSTMIIMSEL is held High to indicate an MDIO transaction.
- HOSTOPCODE maps to the OPCODE field of the MDIO frame.
- HOSTADDR maps to the two address fields of the MDIO frame. PHYAD (the physical address) is HOSTADDR[9:5], and REGAD (the register address) is HOSTADDR[4:0].
- HOSTWRDATA[15:0] maps into the data field of the MDIO frame during a write operation.
- The data field of the MDIO frame maps into HOSTRDDATA[15:0] during a read operation.

A read or write transaction on the MDIO is initiated by a pulse on the HOSTREQ signal. This pulse is ignored if the MDIO interface already has a transaction in progress. The Ethernet MAC deasserts the HOSTMIIMRDY signal while the transaction across the MDIO is in progress. When the transaction across the MDIO interface is completed, the HOSTMIIMRDY signal is asserted by the Ethernet MAC. If the transaction is a read, the data is also available on the HOSTRDATA[15:0] bus.

As noted in Figure 4-7, if a read transaction is initiated, the HOSTRDATA bus is valid at the point indicated. If a write transaction is initiated, the HOSTWRDATA bus must be valid at the point indicated. The bus is locked until the transfer completes.

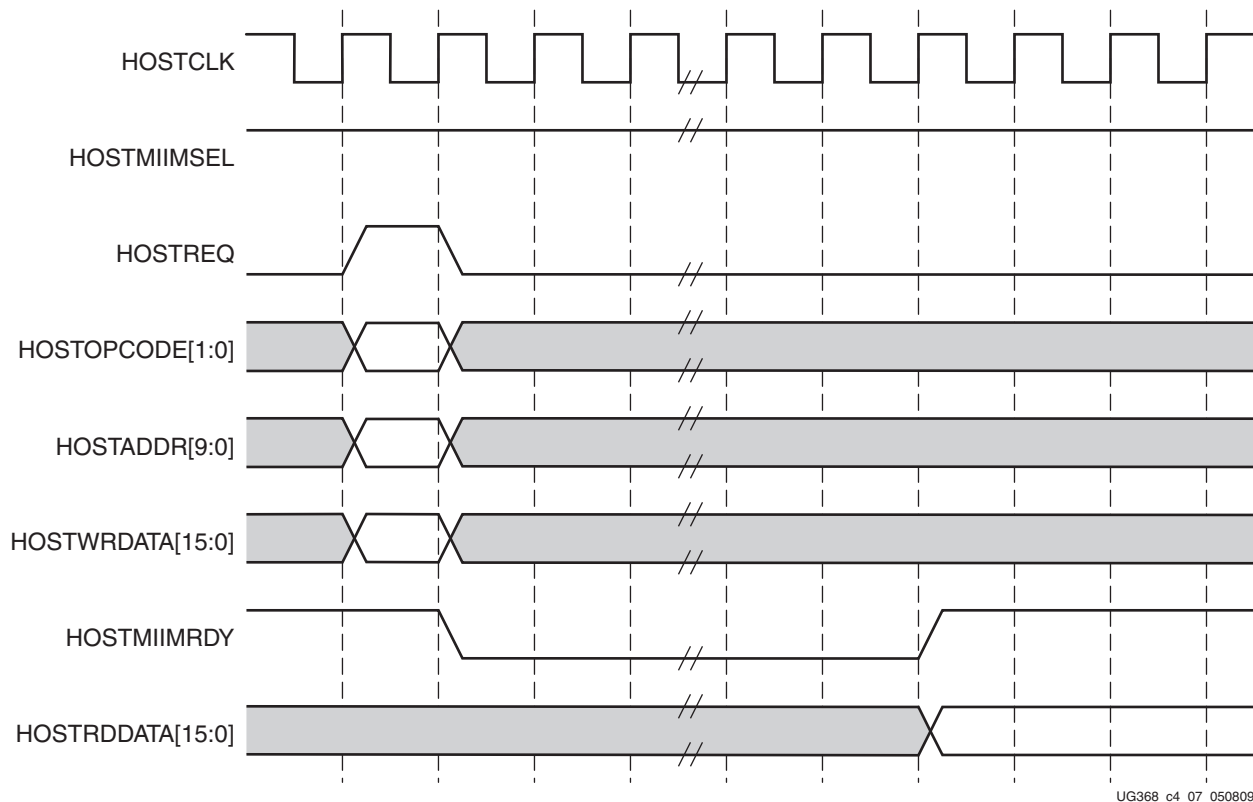


Figure 4-7: MDIO Access through the Host Bus

# Using the DCR Bus

The DCREMACENABLE signal selects between the DCR bus or the host bus during the FPGA power-up configuration. To access the host interface through the DCR bus, the DCREMACENABLE signal must be asserted by tying the signal High in the FPGA logic.

The address space available on the DCR bus is limited. The host interface is accessed indirectly using only four registers in the DCR address space. Accessing the host interface through the DCR bus requires more software complexity than through the host bus. For further information, see the example software routines in [Interfacing to a Processor](#), page 101.

When the DCR bus is used to access the internal registers of the Ethernet MAC, the DCR bus bridge in the host interface translates commands carried over the DCR bus into Ethernet MAC host bus signals. These signals then configure the Ethernet MAC.

## DCR Bus Clocking Requirements

The HOSTCLK frequency must be an integer divide of the DCR clock frequency, and the two clocks must be phase-aligned.

## Device Control Registers

The DCR bus bridge contains four registers directly addressable through the DCR bus for the Ethernet MAC. The upper bits of the DCR address for these registers are user assigned through the EMAC\_DCRBASEADDR attribute.

[Figure 4-8](#) shows how the DCR bridge manages access to the registers within the DCR bus and to the FPGA logic.

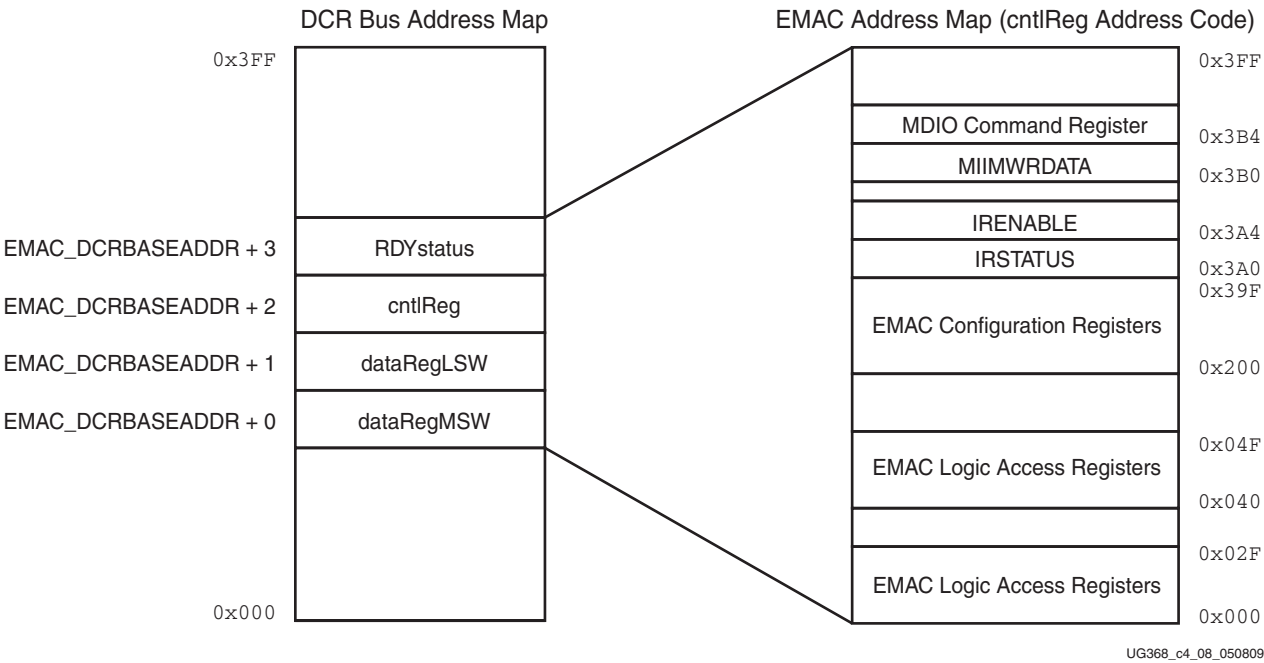


Figure 4-8: DCR Bridge Registers



The directly addressable dataRegMSW, dataRegLSW, cntlReg, and RDYstatus registers, can be read or written to using the appropriate DCR address. By writing to the cntlReg register with a particular address code, the DCR bridge can access the indirect address registers within the DCR bus, such as IRSTATUS, the Ethernet MAC configuration registers, FPGA logic-based registers, or perform an MDIO transaction. Data is read and written to indirect registers through the dataRegLSW register. The dataRegMSW register is also used for transactions with an MSW of data.

Address codes for each transaction type are summarized in [Table 4-15](#).

**Table 4-15: DCR Address Code Transaction Types**

Address Code[9:0]	Transaction Description	Register Accessible Through dataRegLSW
0x000:0x02F	Read from FPGA logic through unused host pins. See <a href="#">Accessing FPGA Logic via Unused Host Bus Pins</a> .	User-defined FPGA logic register.
0x040:0x04F	Read from FPGA logic through unused host pins. See <a href="#">Accessing FPGA Logic via Unused Host Bus Pins</a> .	User-defined FPGA logic register.
0x200:0x390	MAC configuration or address filter register access. See <a href="#">Ethernet MAC Configuration and Address Filter Access</a> .	MAC configuration or address filter register. Address codes correlate to MAC register addresses in <a href="#">Table 4-1</a> .
0x3A0	Interrupt Status. See <a href="#">DCR Interrupts</a> .	IRSTATUS
0x3A4	Interrupt Configuration. See <a href="#">DCR Interrupts</a> .	IRENABLE
0x3B0	MDIO write. See <a href="#">PCS/PMA Sublayer or External Device Access via MDIO</a> .	MIIMWRDATA
0x3B4	MDIO access. See <a href="#">PCS/PMA Sublayer or External Device Access via MDIO</a> .	Registers in PCS/PMA sublayer or from an external device via MDIO.

The register definitions for the DCR bridge registers are shown in Table 4-16 to Table 4-19. The DCR registers (dataRegMSW, dataRegLSW, cntlReg, and RDYstatus) and the DCR bridge registers (IRSTATUS, IRENABLE, and MIIMWRDATA) use the big-endian bit numbering convention. The Ethernet MAC host registers, such as Receiver Configuration (Word 0) (host address 0x200) register, use the little-endian bit numbering convention. In the DCR bridge implementation, there is no conversion to or from big endian to little endian. The bit positions are mapped directly in a one-to-one correspondence (big-endian bit [0] is mapped directly to little-endian bit [31], bit [1] is mapped directly to little-endian bit [30], and so forth).

Table 4-16: DCR Data Register dataRegMSW

MSB	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	LSB
	dataRegMSW																																

Bit	Description	Default Value
[0:31]	Data. Data input from the DCR bus for the Ethernet MAC registers or other accessible registers is written into this register. The most significant word of data is read out from the Ethernet MAC registers or other registers and deposited into this register.	Undefined

The MSW data register is used when reading from the General Address table and for a read from FPGA logic registers. For all other transactions, only the LSW data register is used.

Table 4-17: DCR Data Register dataRegLSW

MSB

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

LSB

dataRegLSW

Bit	Description	Default Value
DCR Data Register (dataRegLSW)		
[0:31]	Data. Data input from the DCR bus for the Ethernet MAC registers or other registers is written into this register. The least significant word of data is read out from the Ethernet MAC registers or other registers is deposited into this register.	Undefined

Table 4-18: DCR Control Register cntlReg

MSB

LSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED																WEN	RESERVED					ADDRESS_CODE									

Bit	Description	Default Value
[0:15]	Reserved.	All 0s
[16]	Write Enable. When this bit is asserted, the data in either dataRegLSW or dataRegMSW is written to the Ethernet MAC host interface. When this bit is deasserted, the operation to be performed is read.	0
[17:21]	Reserved.	All 0s
[22:31]	Address Code. The DCR bus bridge translates the address code for this transaction into the appropriate signals on the Ethernet MAC host interface. See <a href="#">Table 4-15, page 97</a> for a description of address codes.	All 0s

Table 4-19: DCR Ready Status Register RDYstatus (Read Only)

MSB

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

LSB

RESERVED

DCR RDY

RESERVED

CFG WR

CFG RR

AF WR

AF RR

MIIM WR

MIIM RR

FABR RR

Bit	Description	Default Value
[0:14]	Reserved.	All 0s
[15]	DCR Bridge Ready bit. High when all of the bits in RDYstatus are High.	1
[16:24]	Reserved.	All 0s
[25]	Configuration Write-Ready bit.	1
[26]	Configuration Read-Ready bit.	1
[27]	Address Filter Write-Ready bit.	1
[28]	Address Filter Read-Ready bit.	1
[29]	MDIO Write-Ready bit.	1
[30]	MDIO Read-Ready bit.	1
[31]	FPGA Logic Read-Ready bit.	1

The bits in the Ready Status register are set to 1 when there is no access in progress. When an access is in progress, a bit corresponding to the type of access is automatically cleared. The bit is automatically set again when the access is complete.

Table 4-20: Interrupt Status Register IRSTATUS

Address Code	MSB																															LSB
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
0x3A0	RESERVED																								CFG WST	CFG RST	AF WST	AF RST	MIIM WST	MIIM RST	FABR RST	

Bit	Description	Default Value
[0:24]	Reserved.	0
[25]	Configuration Write Interrupt Request Status bit.	0
[26]	Configuration Read Interrupt Request Status bit.	0
[27]	Address Filter Write Interrupt Request Status bit.	0
[28]	Address Filter Read Interrupt Request Status bit.	0
[29]	MDIO Write Interrupt Request Status bit.	0
[30]	MDIO Read Interrupt Request Status bit.	0
[31]	FPGA Logic Read Interrupt Request Status bit.	0

Table 4-21: Interrupt Enable Register IRENABLE

Address Code	MSB																															LSB
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
0x3A4	RESERVED																								CFG WEN	CFG REN	AF WEN	AF REN	MIIM WEN	MIIM REN	FABR REN	

Bit	Description	Default Value
[0:24]	Reserved.	0
[25]	Configuration Write IR-enable bit.	0
[26]	Configuration Read IR-enable bit.	0
[27]	Address Filter Write IR-enable bit.	0
[28]	Address Filter Read IR-enable bit.	0
[29]	MDIO Write IR-enable bit.	0
[30]	MDIO Read IR-enable bit.	0
[31]	FPGA Logic Read IR enable bit.	0

The IRENABLE register is programmed to allow updating of the interrupt request in the IRSTATUS register. When an enable bit is cleared, the corresponding bit in the IRSTATUS register is not updated.

Table 4-22: Host Interface MDIO Write Data Register (MIIMWRDATA)

0x3B0	Address Code	MSB																																LSB
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
		RESERVED																MIIMWRDATA																

Bit	Description	Default Value
[0:15]	Reserved.	All 0s
[16:31]	Data. Temporarily holds MDIO write data for output onto the host write data bus.	Undefined

## Interfacing to a Processor

The DCR bridge ignores any new DCR command for host access until the current host access is complete. Therefore, it is essential to determine when the host access is complete before issuing a new DCR command. While some commands complete within one clock cycle on the DCR bus, others such as MDIO accesses require multiple clock cycles.

There are two ways to use the DCR bus with the host processor. The user can select either polling or interrupts as a method of informing the host processor of access status. The interrupt method is more efficient as it frees the processor to operate while waiting for the DCR transaction to finish.

- Interrupt request informs the host processor when the host interface completes a DCR host access command. The interrupt request signal, DCRHOSTDONEIR, is only active when the DCR bus is used, and the host interface register IRENABLE is programmed to enable interrupt. This signal is active High and level sensitive. When a host access through the DCR bus is completed, the DCRHOSTDONEIR signal is asserted. The host processor needs to service the interrupt request and clear the host interface register (IRSTATUS) to deassert this signal. See [DCR Interrupts](#) for a description of the DCR Interrupts.
- Polling of the DCR status register RDYstatus. The processor polls this register to determine access completion status. The bits in this register are asserted when there is no access in progress. When an access is in progress, a bit corresponding to the type of access is automatically deasserted. This bit is automatically reasserted when the access is complete. RDYstatus [15] is asserted only when there are no DCR transactions in progress, allowing the polling process to check this one bit for all transaction types.

## DCR Interrupts

The DCRHOSTDONEIR interrupt can be used to manage processor DCR access. The operation and status of the interrupt can be accessed through the IRSTATUS and IRENABLE registers.

The IRSTATUS register indicates which transaction type has triggered the interrupt. This allows the host to read this register to determine the type of host access completed. Before

exiting the interrupt service routine, the host should write to this register to clear the interrupt request bit.

To read and clear the IRSTATUS register, the following operations must be performed on the DCR bus:

1. To access the IRSTATUS register, write an address code of 0x3A0 to the cntlReg register, addressing cntlReg using EMAC\_DCRBASEADDR for IRSTATUS. The write enable bit should be cleared to indicate a read.
2. The register information for IRSTATUS is loaded into dataRegLSW. Read this register to access the data.
3. To clear the interrupt, the contents of the IRSTATUS register should be written to zero. Write a value of all zeros into the dataRegLSW.
4. Write the cntlReg with the address 0x3A0 to load the data from dataRegLSW into the register IRSTATUS and clear the interrupt. The write enable bit should be set to 1 to indicate a write.

To read from the IRSTATUS register and then clear the register:

```
// Write the address of IRStatus register 0x3A0 to the
// cntlReg register. Ensure that the write enable bit is not asserted
dcr_write(EMAC_DCRBASEADDR + 2, 0x000003A0);
// Read the IRSTATUS data from the dataRegLSW
dcr_read (EMAC_DCRBASEADDR + 1);
// Write a value of all zeros to the dataRegLSW
dcr_write(EMAC_DCRBASEADDR + 1, 0x00000000);
// Write to the cntlReg register to load the data into the
// IRStatus register to clear the interrupt
dcr_write(EMAC_DCRBASEADDR + 2, 0x000083A0);
```

The IRENABLE register is programmed to enable the different interrupt types to trigger the DCRHOSTDONEIR interrupt. When an enable bit is set, the corresponding bit in the IRSTATUS register is updated when an interrupt of that type occurs. To read from this register and then write back, the following steps are required.

1. To read the IRENABLE register, write an address code 0x3A4 to the cntlReg, addressing the cntlReg using EMAC\_DCRBASEADDR for IRENABLE, and the write enable bit should be cleared to indicate a read operation.
2. The register information for IRENABLE is loaded into the dataRegLSW register. Read this register to access the data.
3. To enable specific interrupts, write the required bitmask value to the dataRegLSW register.
4. Write to the cntlReg with the address code 0x3A4 to load the data from datRegLSW into the IRENABLE register. The write enable bit should be set to 1 to indicate a write operation.

To read from the IRENABLE register and then enable all interrupts:

```
// Write the address of IRENABLE register 0x3A4 to the
// cntlReg register. Ensure that the write enable bit is not asserted
dcr_write(EMAC_DCRBASEADDR + 2, 0x000003A4);
// Read the IRENABLE data from the dataRegLSW
dcr_read (EMAC_DCRBASEADDR + 1);
// Write a value of all ones to the dataRegLSW
dcr_write(EMAC_DCRBASEADDR + 1, 0x0000007F);
// Write to the cntlReg register to load the data into the
// IRENABLE register and enable all interrupts
dcr_write(EMAC_DCRBASEADDR + 2, 0x000083A4);
```

## Ethernet MAC Configuration and Address Filter Access

### Reading from an Ethernet MAC Configuration Register

To read from an Ethernet MAC configuration register, the user must perform the following sequence of operations on the DCR bus:

1. Using EMAC\_DCRBASEADDR, write to the cntlReg register, with the write enable bit cleared to 0, to indicate a read operation and set the address code to the desired address of the Ethernet MAC configuration register to be accessed. The address code maps directly to the Ethernet MAC address codes given in [Table 4-1](#).
2. Poll the RDYstatus register until the configuration Read-Ready bit is set to 1 or wait until the DCRHOSTDONEIR interrupt is asserted.
3. Read from the dataRegLSW register to access data from the Ethernet MAC configuration register.

The following code demonstrates a processor routine to read from an Ethernet MAC configuration register. To read from the transmitter configuration register:

```
// EMAC Configuration Register 0x280 (Transmitter Configuration)
// Write the address of Transmitter Configuration register to the
// cntlReg register
dcr_write(EMAC_DCRBASEADDR + 2, 0x00000280);
// Poll the RDYstatus register
while ( !(dcr_read(EMAC_DCRBASEADDR + 3) & 0x00010000) );
// Read the dataRegLSW with the values returned from the
// Transmitter Configuration register
dcr_read (EMAC_DCRBASEADDR + 1);
```

### Writing to an Ethernet MAC Configuration Register

To write to an Ethernet MAC configuration register, the user must perform the following sequence of operations on the DCR bus:

1. Write to the dataRegLSW register with the desired value for the Ethernet MAC configuration register. This value is mapped to the host configuration register.
2. Using EMAC\_DCRBASEADDR, write to the cntlReg register with the desired address of the Ethernet MAC configuration register and the write enable bit to logic 1.
3. Poll the RDYstatus register until the configuration Write-Ready bit is set to 1 or wait until the DCRHOSTDONEIR interrupt is asserted.

To write to the flow control register:

```
// EMAC Configuration Register 0x2C0 (EMAC Flow Control)
// Write to enable the flow control on both the transmit and receive
// side, set bits 29 and 30 to 1
dcr_write(EMAC_DCRBASEADDR + 1, 0x60000000);
// Write the address of Flow Control register to the cntlReg
// register, with the write enable bit asserted
dcr_write(EMAC_DCRBASEADDR + 2, 0x000082C0);
// Poll the RDYstatus register
while ( !(dcr_read(EMAC_DCRBASEADDR + 3) & 0x00010000) );
```

## Reading from the General Address Table Register of the Address Filter Block

The same methods used in reading and writing to the Ethernet MAC configuration registers through the DCR apply to the address filter configuration registers. The only operations that differ are reading and writing from the general address table.

To read the desired general address table register of the address filter, a DCR write operation must be performed.

1. Write to the dataRegLSW register, setting the two-bit address of the general address table of the register to be accessed (four general address table registers are in the address filter block) and the RNW bit to 1.
2. Using EMAC\_DCRBASEADDR, write to the cntlReg register with the address register of general address (Word 1) to 0x38C. Set the Write enable bit to write to the general address (Word 1) (see [Table 4-12](#)).
3. Poll the RDYstatus register until the address filter Read-Ready bit is set to 1 or wait until the DCRHOSTDONEIR interrupt is asserted.
4. Read from the dataRegMSW to read the general address [47:32] and dataRegLSW to read general address [31:0].

To read from the general address table register (2):

```
// MULTI_ADDR Register 2 of AF Block
// Set the enable bit of the MULTI_ADDR RNW and MULTI_ADDR Register 2
dcr_write(EMAC_DCRBASEADDR + 1, 0x00820000);
// Write the address of General Address register to the cntlReg
// register, with the write enable bit asserted
dcr_write(EMAC_DCRBASEADDR + 2, 0x0000838C);
// Poll the RDYstatus register
while ( !(dcr_read(EMAC_DCRBASEADDR + 3) & 0x00010000) );
// Read the values returned of the General Address Word0 and Word1
// registers (48-bit value)
// from the dataRegMSW and dataRegLSW registers
gen_addr_msw = dcr_read(EMAC_DCRBASEADDR + 0);
gen_addr_lsw = dcr_read(EMAC_DCRBASEADDR + 1);
```

## Writing to the General Address Table Register of the Address Filter Block

The same methods used in reading and writing to the Ethernet MAC configuration registers through the DCR apply to the address filter configuration registers. The only operations that differ are reading and writing from the general address table.



For writing to the desired general address table register of the AF block, the following must be performed:

1. Write to the dataRegLSW register the general address [31:0] to be stored on the desired general address table register.
2. Using EMAC\_DCRBASEADDR, write to the cntlReg register with the address for general address word 0 (0x388), setting the write enable bit.
3. Poll the RDYstatus register until the address configuration write bit is set to 1.
4. Write to the dataRegLSW register the general address [47:32] with the write mask bit and the value of the general address table register to be accessed.
5. Write to cntlReg register with the address for general address word 1 (0x38C). Set the write enable bit.
6. Poll the RDYstatus register until the address configuration write bit is set or wait until the DCRHOSTDONEIR interrupt is asserted.

To write the general address 0xFACEDAEAFCAFE to the general address table register 0x1:

```
// Write the general address[31:0] to the dataRegLSW register
dcr_write(EMAC_DCRBASEADDR + 1, 0xDEAFCAFE);
// Write the address of the General Address Word 0 register to the
// cntlReg register
dcr_write(EMAC_DCRBASEADDR + 2, 0x00008388);
// Poll the RDYstatus register
while ( !(dcr_read(EMAC_DCRBASEADDR + 3) & 0x00010000) );
// MULTI_ADDR Register 1 of AF Block
// Write the general address [47:32] with the MULTI_ADDR RNW
// write mask bit deasserted to the dataRegLSW register
dcr_write(EMAC_DCRBASEADDR + 1, 0x0001FACE);
// Write the address of the General Address Word 1 register to the
// cntlReg register
dcr_write(EMAC_DCRBASEADDR + 2, 0x0000838C);
// Poll the RDYstatus register
while ( !(dcr_read(EMAC_DCRBASEADDR + 3) & 0x00010000) );
```

## PCS/PMA Sublayer or External Device Access via MDIO

MDIO register reads take a multiple number of host clock cycles depending on the physical interface device. To write to any of the PCS layer registers ([1000BASE-X PCS/PMA Management Registers, page 118](#)), the data must be written to the MIIMWRDATA register. The physical address (PHYAD) and PCS register number (REGAD) are written to the DCR dataRegLSW register for both read and write operations. Writing the address code 0x3B4 to the control register indicates a request for an MDIO transaction. The format for writing the PHYAD and REGAD into the dataRegLSW register is shown in [Figure 4-9](#). To access the internal PCS/PMA sublayer registers, the PHYAD should match that set on the PHYEMACPHYAD[4:0] input port. The MDIO interface is described in detail in [Chapter 5](#).

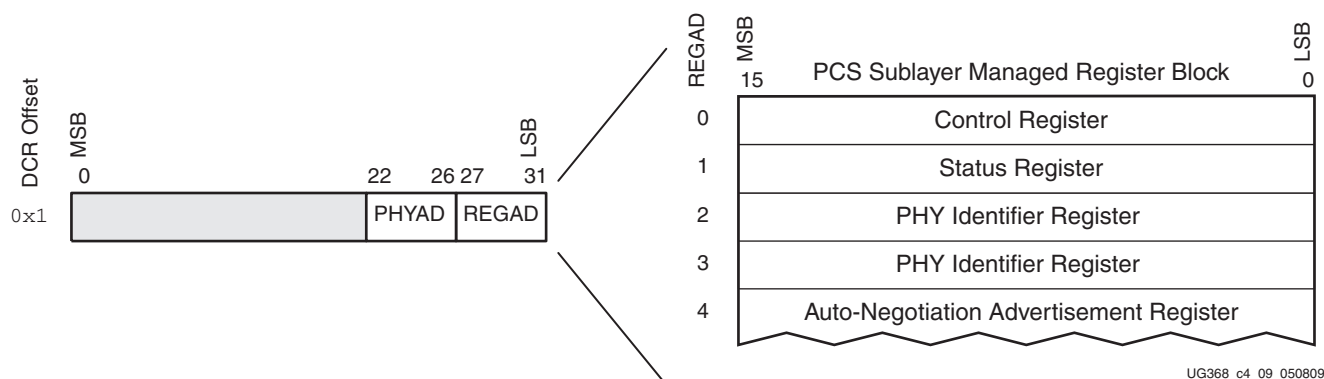


Figure 4-9: MDIO Address Register to Access PCS Sublayer Register Block

### Reading the PHY Registers Using MDIO

To read from a configuration register either in the Ethernet MAC PCS/PMA layer or from an external PHY, the user must perform the following sequence of operations on the DCR bus:

1. Write to the dataRegLSW register with the PHYAD and register to be accessed. This value should be formatted as shown in [Figure 4-9](#).
2. Using EMAC\_DCRBASEADDR, write to the cntlReg register with the decode address for MDIO transaction (0x3B4) and the write enable bit cleared to indicate a read operation.
3. Poll the RDYstatus register until the MDIO Write-Ready bit is set or wait until the DCRHOSTDONEIR interrupt is asserted.
4. Read the PHY register contents from the dataRegLSW.

To read PHYAD 0x1 and PHY register 0x4 through the MDIO interface, MDIO must be enabled by writing to the management configuration register with the clock divider for MDC, as described in [Writing to an Ethernet MAC Configuration Register](#).

```
// Write the PHY address and PHY register to be accessed to the
// dataRegLSW register
dcr_write(EMAC_DCRBASEADDR + 1, (0x1 << 5) + 0x4);
// Write the decode address for MDIO address output to the cntlReg
// register
dcr_write(EMAC_DCRBASEADDR + 2, 0x000003B4);
// Poll the RDYstatus register
while ( !(dcr_read(EMAC_DCRBASEADDR + 3) & 0x00010000) );
// Read the PHY register contents from dataRegLSW
dcr_read(EMAC_DCRBASEADDR + 1);
```

## Writing to the PHY Registers Using MDIO

The MIIMWRDATA register temporarily holds MDIO write data for output to the MDIO write data bus. To write to a configuration register either in the Ethernet MAC PCS/PMA layer or to an external PHY, the user must perform the following sequence of operations on the DCR bus.

1. Write to the dataRegLSW register with the data to be written to the PHY register.
2. Write to the cntlReg register with the address for the MDIO write data register (0x3B0). This operation loads the write data from dataRegLSW into the MIIMWRDATA register.
3. Write to the dataRegLSW register with the PHY address and register to be accessed. This operation should be formatted as shown in [Figure 4-9](#).
4. Using EMAC\_DCRBASEADDR, write to the cntlReg register the decode address for MDIO transaction (0x3B4) with the write enable bit set to indicate a write operation.
5. Poll the RDYstatus register until the MDIO Write-Ready bit is set or wait until the DCRHOSTDONEIR interrupt is asserted.

To write 0x1140 to PHYAD 0x2 and PHY register 0x0, MDIO must be enabled by writing to the management configuration register with the clock divider for MDC, as described in [Writing to an Ethernet MAC Configuration Register](#).

```
// EMAC Management Register 0x340 (EMAC Management Configuration)
// Write the PHY data to the dataRegLSW
dcr_write(EMAC_DCRBASEADDR + 1, 0x00001140);
// Write the address for MIIMWRDATA to the cntlReg
// register to load the write data to MIIMWRDATA
dcr_write(EMAC_DCRBASEADDR + 2, 0x000083B0);
// Write the PHY address and PHY register to be accessed to the
// dataRegLSW register
dcr_write(EMAC_DCRBASEADDR + 1, (0x2 << 5) + 0x0);
// Write the decode address for MDIO address output to the cntlReg
// register to initiate the transaction
dcr_write(EMAC_DCRBASEADDR + 2, 0x000083B4);
// Poll the RDYstatus register
while ( !(dcr_read(EMAC_DCRBASEADDR + 3) & 0x00010000) );
```

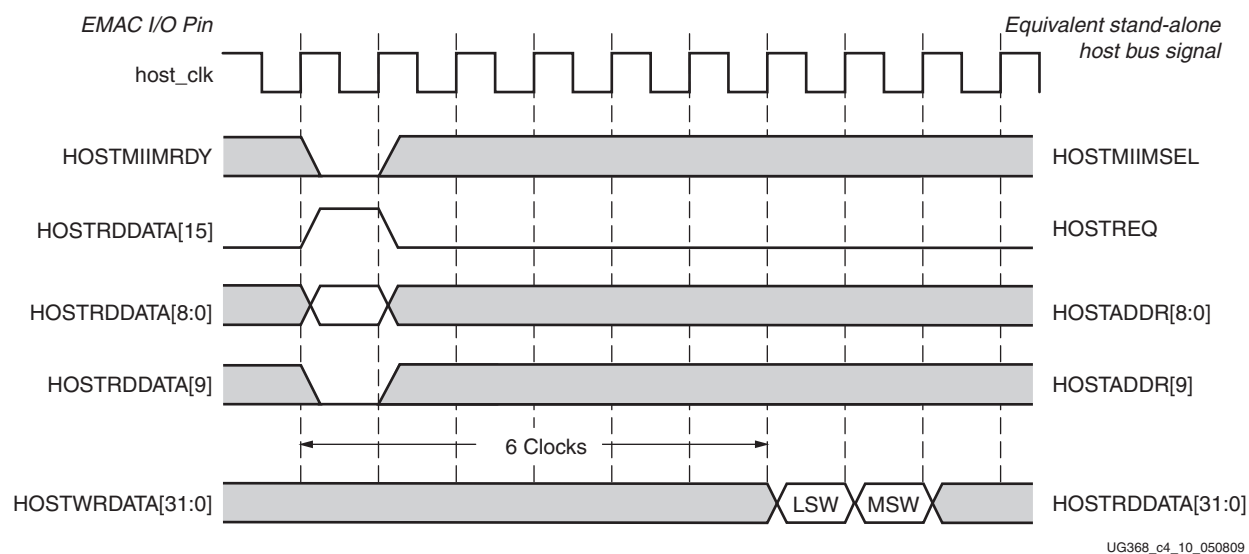
## Accessing FPGA Logic via Unused Host Bus Pins

The host bus I/O pins of the Ethernet MAC are enabled for access to the FPGA logic when a DCR read operation is made for address codes 0x000 to 0x02F and 0x040 to 0x04F inclusive ([Table 4-15](#) describes the DCR address code space). This use of the host bus I/O signals provides a means of accessing the FPGA logic from the processor with space for 64 addresses for each EMAC.

When the DCR bus is instructed to access registers in this address code region, the DCR bridge translates the DCR commands into generic host read signals on the host bus I/O signals. The DCR transaction is encoded on the host bus signals HOSTRDDATA[31:0] and HOSTMIIMRDY as described in [Table 4-23](#) and [Figure 4-10](#). These signals can be used to access an FPGA logic-based Statistics block or other IP. Data can be read in from the logic from HOSTWRDATA[31:0], as shown in [Figure 4-10](#). The data read from the host bus can then be accessed by reading the DCR data registers.

Table 4-23: DCR to Host Mapping

EMAC Host Bus Port		Port Direction on EMAC	DCR Register Decode	Equivalent Stand-alone Host Bus Signal
HOSTRDDATA	[15]	Out		HOSTREQ
	[14:13]	Out		HOSTOPCODE[1:0]
	[9:0]	Out	cntlReg[22:31], Address Code	HOSTADDR[9:0]
HOSTMIIMRDY		Out		HOSTMIIMSEL
HOSTWRDATA[31:0]		In	dataRegLSW / dataRegMSW	HOSRTRDDATA[31:0]
HOSTMIIMSEL		In		HOSTMIIMRDY



UG368\_c4\_10\_050809

Figure 4-10: FPGA-Logic Based Register Read Timing

To read from the FPGA logic, the following operations should take place on the DCR bus.

1. Write to the `cntlReg` register with the address code in the range `0x000` to `0x02F` or `0x040` to `0x04F`. This address is encoded onto the `HOSTRDDATA[9:0]` I/O pins. The write enable bit is cleared to indicate a read operation.
2. Poll the `RDYstatus` register until the Statistics IP Read-Ready bit is set or wait until the `DCRHOSTDONEIR` interrupt is asserted.
3. Read the `dataRegMSW` and `dataRegLSW` registers to get the contents of the FPGA logic-based registers.

To read a logic-based register addressed 0x020:

```
// FPGA Logic Register at 0x020 (User defined)
// Write the decode address for FPGA logic access to the cntlReg
// register
dcr_write(EMAC_DCRBASEADDR + 2, 0x00000020);
// Poll the RDYstatus register
while ( !(dcr_read(EMAC_DCRBASEADDR + 3) & 0x00010000) );
// Read the value from MSW and LSW registers
fabric_msw = dcr_read(EMAC_DCRBASEADDR + 0);
fabric_lsw = dcr_read(EMAC_DCRBASEADDR + 1);
```

This method can be used to connect the Ethernet MAC to the statistics counters as described in [Chapter 7, Interfacing to a Statistics Block](#).



## MDIO Interface

---

This chapter describes the Management Data Input/Output (MDIO) interface that is used to access physical device registers, both external to the Ethernet MAC (for example an external PHY device) and internal to the Ethernet MAC (the physical device registers present in the PCS/PMA sublayer block).

The internal physical device (i.e., the PCS/PMA sublayer block) is capable of 1000BASE-X PCS/PMA and SGMII operation (see [Chapter 6, Physical Interface](#)). These two modes of operation each have associated management registers.

This chapter contains the following sections:

- [Introduction to MDIO](#)
- [MDIO Implementation in the Ethernet MAC](#)
- [1000BASE-X PCS/PMA Management Registers](#)
- [SGMII Management Registers](#)

# Introduction to MDIO

## MDIO Bus System

The MDIO interface for 1 Gb/s operation (and slower speeds) is defined in IEEE Std 802.3-2005, clause 22. This two-wire interface consists of a clock (MDC) and a shared serial data line (MDIO). The maximum permitted frequency of MDC is set at 2.5 MHz.

Figure 5-1 illustrates an example MDIO bus system.

- An Ethernet MAC is shown as the MDIO bus master (the Station Management (STA) entity).
- Two PHY devices are shown connected to the same bus, both are MDIO slaves (MDIO Managed Device (MMD) entities).

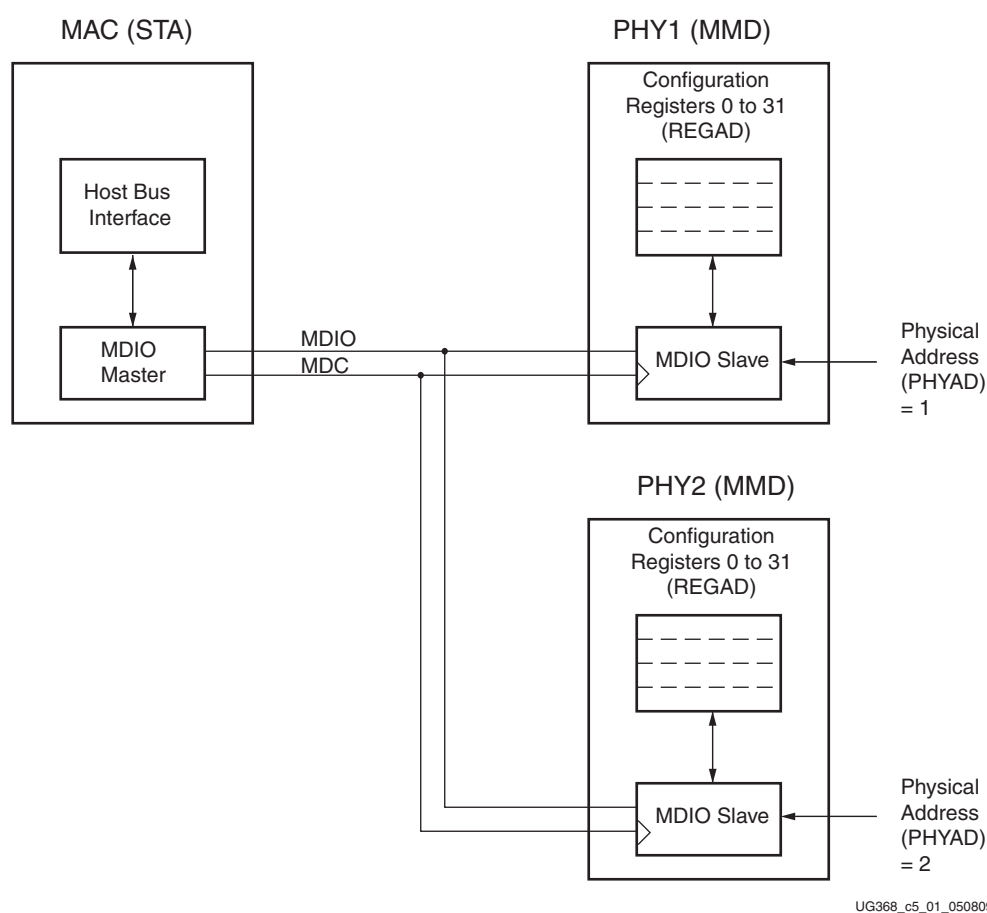


Figure 5-1: Typical MDIO-Managed System

Described simply, the MDIO bus system is a standardized interface for accessing the configuration and status registers of Ethernet PHY devices. In the example illustrated, the Management Host Bus Interface of the Ethernet MAC is able to access the configuration and status registers of two PHY devices via the MDIO bus.



## MDIO Transactions

All transactions, read or write, are initiated by the MDIO master. All MDIO slave devices, when addressed, must respond. MDIO transactions take the form of an MDIO frame, containing fields for transaction type, address and data. This MDIO frame is transferred across the MDIO wire synchronously to MDC. The following abbreviations are used in this chapter:

Table 5-1: Abbreviations Used in this Chapter

Abbreviation	Definition
OP	Operation code (read or write)
PHYAD	Physical address
PRE	Preamble
REGAD	Register address
ST	Start of frame
TA	Turnaround

### Write Transaction

Figure 5-2 shows a write transaction frame across the MDIO, as defined by OP = 0b01. The addressed PHY device (with physical address PHYAD) takes the 16-bit word in the Data field and writes it to the register at REGAD.

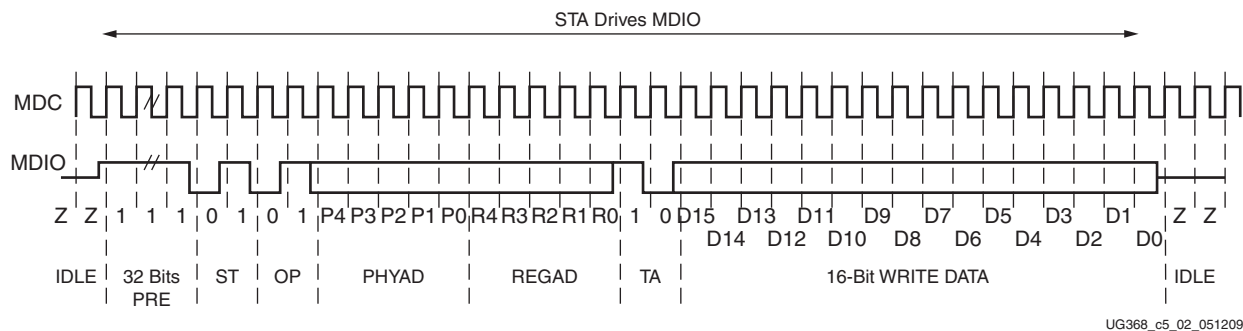


Figure 5-2: MDIO Write Transaction

### Read Transaction

Figure 5-3 shows a read transaction as defined by OP = 0b10. The addressed PHY device (with physical address PHYAD) takes control of the MDIO wire during the turnaround cycle and then returns the 16-bit word from the register at REGAD.

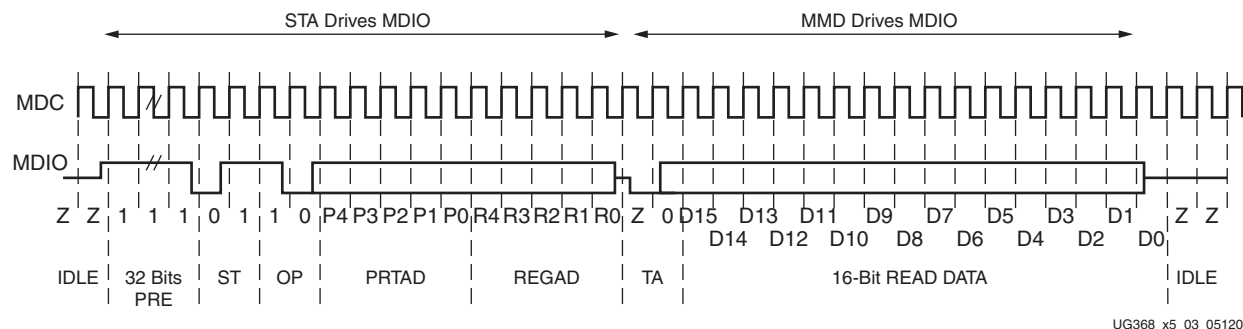


Figure 5-3: MDIO Read Transaction

## MDIO Addressing

MDIO addresses consists of two stages: physical address (PHYAD) and register address (REGAD).

### Physical Address (PHYAD)

As shown in [Figure 5-1](#), two PHY devices are attached to the MDIO bus. Each of these has a different physical address. To address the intended PHY, its physical address should be known by the MDIO master (in this case an Ethernet MAC) and placed into the PHYAD field of the MDIO frame (see [MDIO Transactions](#)).

The PHYAD field for an MDIO frame is a 5-bit binary value capable of addressing 32 unique addresses. However, every MDIO slave must respond to physical address 0. This requirement dictates that the physical address for any particular PHY must not be set to 0 to avoid MDIO contention. Physical Addresses 1 through to 31, therefore, can be used to connect up to 31 PHY devices onto a single MDIO bus.

Physical Address 0 can be used to write a single command that is obeyed by all attached PHYs, such as a reset or power-down command.

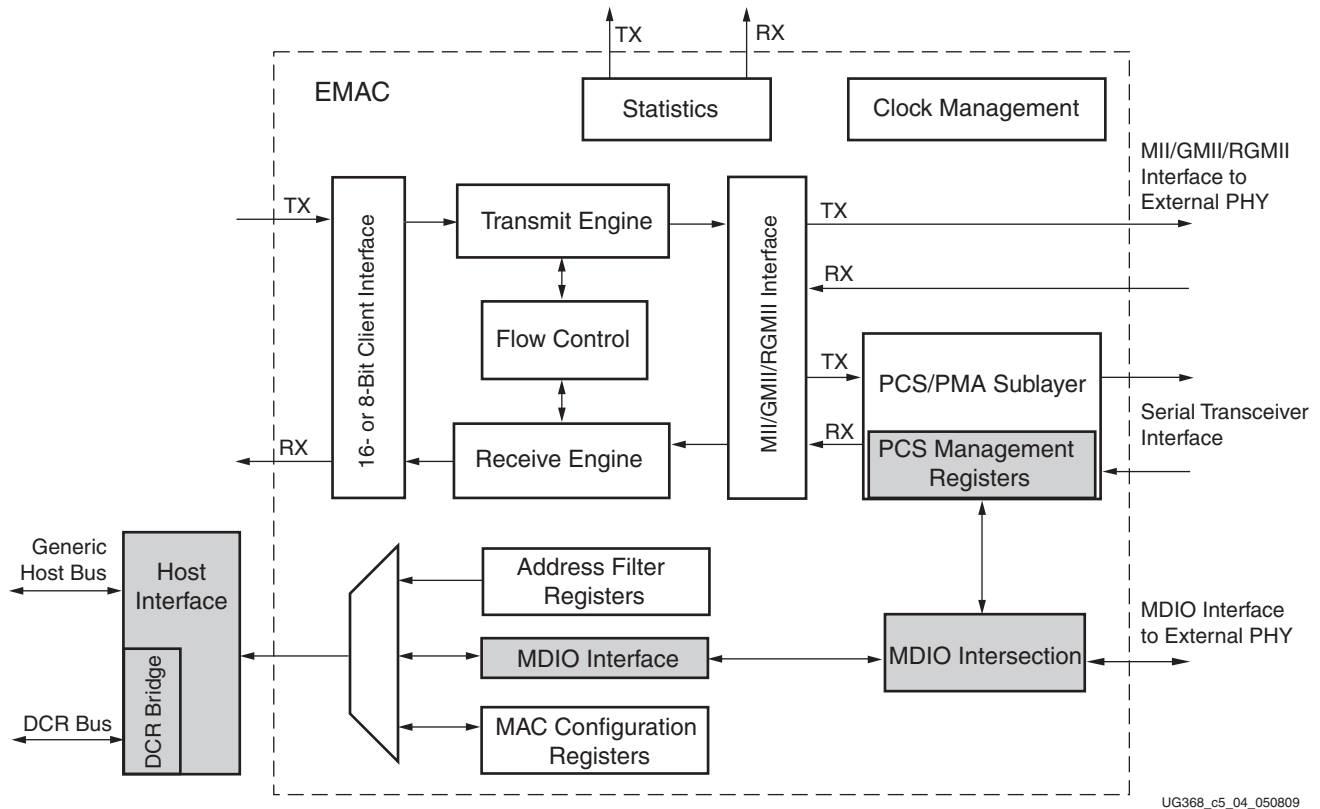
### Register Address (REGAD)

Having targeted a particular PHY using PHYAD, the individual configuration or status register within that particular PHY must now be addressed by placing the individual register address into the REGAD field of the MDIO frame (see [MDIO Transactions](#)).

The REGAD field for an MDIO frame is a 5-bit binary value capable of addressing 32 unique addresses. The first 16 of these registers (0 to 15) are defined by the IEEE Std 802.3-2005. The remaining 16 registers (16 to 31) are reserved for PHY vendors' own register definitions.

## MDIO Implementation in the Ethernet MAC

Figure 5-4 shows the functional block diagram of the Ethernet MAC with the host interface, DCR bridge, MDIO interface, PCS management, and MDIO intersection blocks highlighted.



**Figure 5-4: MDIO Implementation of the Ethernet MAC**

## Accessing MDIO through the Host Interface

The Ethernet MAC contains an MDIO interface block which is an MDIO master. This can initiate MDIO transactions when accessed through either the generic host bus or the DCR bus. See [PCS/PMA Sublayer or External Device Access via MDIO, page 94](#), for reference.

The MDIO interface supplies a clock, EMACPHYMCLKOUT, to the external devices when the host interface is enabled. This clock is derived from the HOSTCLK signal using the value in the Clock Divide[5:0] configuration register. The frequency of the MDIO clock is given by [Equation 5-1](#):

$$f_{MDC} = \frac{f_{HOSTCLK}}{((1 + \text{CLOCK\_DIVIDE}[5:0]) \times 2)} \quad \text{Equation 5-1}$$

To comply with the IEEE Std 802.3-2005 specification for this interface, the frequency of EMACPHYMCLKOUT should not exceed 2.5 MHz. To prevent EMACPHYMCLKOUT from being out of specification, the Clock Divide[5:0] value powers up at 000000. While this value is in the register, it is impossible to enable the MDIO interface. Upon reset, the MDIO port is disabled until a non-zero value has been written into the clock divide bits.

From the MDIO Interface block, the MDIO bus is internally connected within the Ethernet MAC to the MDIO Intersection block. Here the MDIO bus is split. It is internally connected

to the 1000BASE-X PCS/PMA sublayer logic and connected to the Ethernet MAC MDIO ports (see [Management Data Input/Output \(MDIO\) Signals, page 32](#)), which can provide MDIO access to and from an external PHY device. See [Table 2-16, page 37](#). The EMAC\_MDIO\_ENABLE attribute must be set (or MDIO Enable must be set in the Management Configuration Register) to allow access to the internal MDIO registers and the external MDIO bus.

## Accessing the Internal PCS/PMA Sublayer Management Registers

The PCS/PMA sublayer logic, illustrated as a block in [Figure 5-4](#), contains an MDIO slave, allowing access to its configuration and status registers. All connections are internal and active whenever MDIO operation is enabled.

The PHYAD of the internal PCS/PMA sublayer registers can be set with the PHYEMACPHYAD[4:0] port. To access the PCS/PMA sublayer registers, simply initiate an MDIO transaction using the matching physical address. The PCS/PMA sublayer also responds to Physical Address 0 (see [MDIO Addressing, page 114](#)), unless the EMAC\_MDIO\_IGNORE\_PHYADZERO parameter is set to TRUE (see [Table 2-18](#)).

The PCS/PMA Management registers have different definitions depending on the mode of operation. See as appropriate:

- [1000BASE-X PCS/PMA Management Registers](#)
- [SGMII Management Registers](#)

When no external PHY is present, the PHYEMACMDIN port should be tied High and the PHYEMACMCLKIN port tied Low.

## Accessing the Management Registers of an External PHY using MDIO

Whenever an MDIO operation is enabled (see [Table 2-16, page 37](#)), connections can be made to the Ethernet MAC MDIO ports (see [Management Data Input/Output \(MDIO\) Signals, page 32](#)), which can provide MDIO access to and from an external PHY device.

PHYEMACMDIN, EMACPHYMDOUT, and EMACPHYMDTRI must be connected to a three-state buffer to create the bidirectional wire, MDIO. This three-state buffer can be either external to the FPGA or internally integrated using an IOBUF with an appropriate I/O standard for the external PHY. This is illustrated in [Figure 5-5](#).

In this mode of operation, the MDC clock is generated by the Ethernet MAC and obtained from the output port EMACPHYMCLKOUT; the input PHYEMACMCLKIN port is unused and should be connected to logic 0.

To access the external PHY registers, simply initiate an MDIO transaction using the appropriate physical address. The PCS/PMA sublayer, even if it is not the physical interface, is still connected and responds to physical addresses 0 and the address matching the value of the PHYEMACPHYAD[4:0] port.

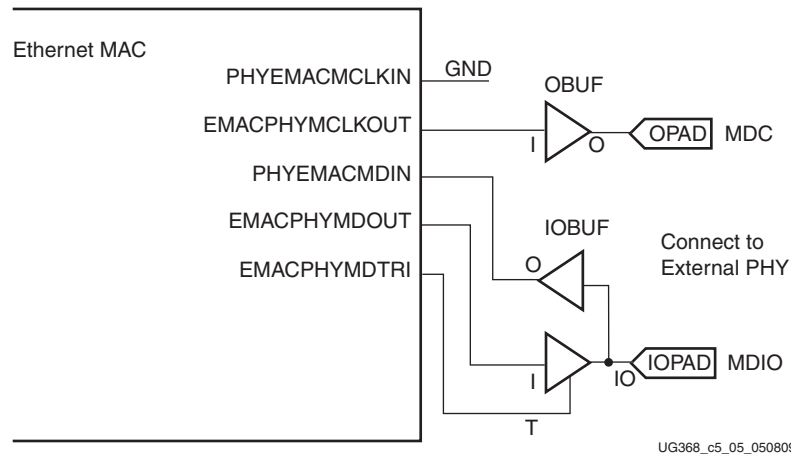


Figure 5-5: MDIO Access to External PHY

## Accessing PCS/PMA Sublayer Management Registers using MDIO

Figure 5-6 shows the functional block diagram of the Ethernet MAC with the PCS management register, MDIO intersection, and MDIO master blocks highlighted. This figure shows that the PCS/PMA sublayer registers can still be accessed via MDIO when the host interface is not in use.

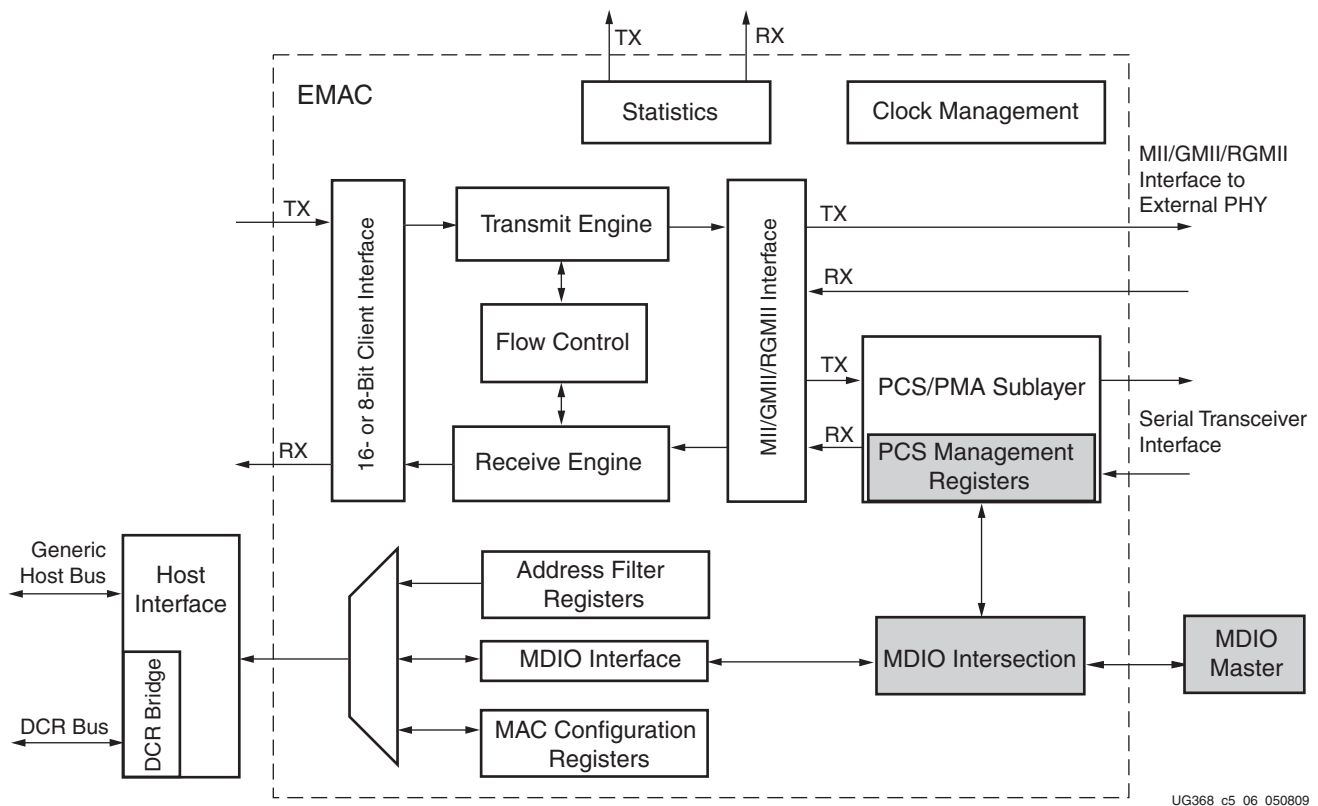


Figure 5-6: Accessing PCS/PMA Sublayer Management Registers in the Ethernet MAC

To enable this functionality, the host interface should be disabled, and MDIO operation should be enabled (see [Table 2-16, page 37](#)).

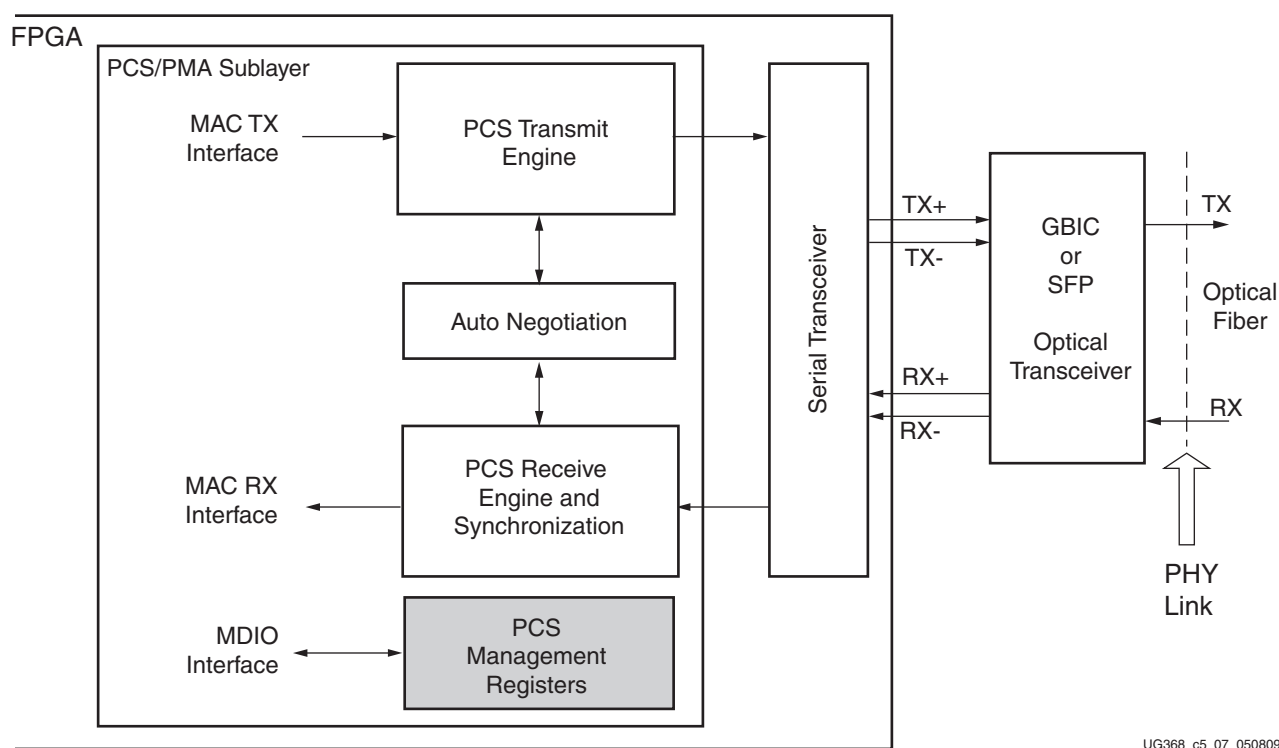
In this situation, an MDIO master external to the Ethernet MAC must initiate the [MDIO Transactions](#). This MDIO master can exist in FPGA logic or as an external device. In this situation, the MDC clock must be provided to the Ethernet MAC through the PHYEMACMCKIN port. The output signal EMACPHYMCLKOUT should be left unconnected.

The PCS/PMA Management registers have different definitions depending on the mode of operation. See as appropriate:

- 1000BASE-X PCS/PMA Management Registers
- SGMII Management Registers

## 1000BASE-X PCS/PMA Management Registers

The PCS/PMA sublayer block, shown in [Figure 5-7](#), is contained within the Ethernet MAC. The PCS/PMA sublayer contains PCS Management registers, as listed in [Table 5-2](#). These registers are described in further detail in [Table 5-3](#) through to [Table 5-13](#).



**Figure 5-7: 1000BASE-X PCS/PMA Sublayer Logic and Physical Connections**

Registers 0 through to 15, shown in [Figure 5-7](#), are defined in IEEE Std 802.3-2005. These contain information relating to the operation of the 1000BASE-X PCS/PMA sublayer, including the status of the physical Ethernet link (PHY link). Additionally, these registers are directly involved in the operation of the 1000BASE-X auto-negotiation function that occurs between the Ethernet MAC and its link partner, which is the Ethernet device connected at the far end of the PHY link (see [1000BASE-X Auto-Negotiation](#), [page 164](#)).

Registers 16 and 17 are vendor-specific registers, defined by Xilinx.

Table 5-2: Configuration Registers for 1000BASE-X PCS/PMA

Register Address (REGAD)	Register Name
0	Control Register (Register 0)
1	Status Register (Register 1)
2,3	PHY Identifier (Registers 2 and 3)
4	Auto-Negotiation Advertisement Register (Register 4)
5	Auto-Negotiation Link Partner Ability Base Register (Register 5)
6	Auto-Negotiation Expansion Register (Register 6)
7	Auto-Negotiation Next Page Transmit Register (Register 7)
8	Auto-Negotiation Next Page Receive Register (Register 8)
15	Extended Status Register (Register 15)
16	Vendor-Specific Register: Auto-Negotiation Interrupt Control Register (Register 16)
17	Vendor-Specific Register: Loopback Control Register (Register 17)

Table 5-3: Control Register (Register 0)

Bit(s)	Name	Description	Type	Default Value
0.15	Reset	1 = PCS/PMA reset. 0 = Normal operation.	Read/Write Self Clearing	EMAC_PHYRESET See Table 2-18, page 41.
0.14	Loopback	1 = Enable loopback mode. 0 = Disable loopback mode.	Read/Write	EMAC_PHYLOOPBACKMSB See Table 2-18, page 41.
0.13	Speed Selection (LSB)	The Ethernet MAC always returns a 0 for this bit. Along with bit 0.6, speed selection of 1000 Mb/s is identified.	Returns 0	0
0.12	Auto-Negotiation Enable	1 = Enable auto-negotiation process. 0 = Disable auto-negotiation process.	Read/Write	EMAC_PHYINITAUTONEG_ENABLE See Table 2-18, page 41.
0.11	Power Down	1 = Power down. 0 = Normal operation. When set to 1, the serial transceiver is placed in a Low power state. This bit requires a reset (see bit 0.15) to clear.	Read/ Write	EMAC_PHYPOWERDOWN See Table 2-18, page 41.
0.10	Isolate	1 = Electrically isolate the PHY from GMII. 0 = Normal operation.	Read/Write	EMAC_PHYISOLATE See Table 2-18, page 41.
0.9	Restart Auto-Negotiation	1 = Restart auto-negotiation process. 0 = Normal operation.	Read/Write Self Clearing	0

Table 5-3: Control Register (Register 0) (Cont'd)

Bit(s)	Name	Description	Type	Default Value
0.8	Duplex Mode	The Ethernet MAC always returns a 1 for this bit to signal full-duplex mode.	Returns 1	1
0.7	Collision Test	The Ethernet MAC always returns a 0 for this bit to disable COL test.	Returns 0	0
0.6	Speed Selection (MSB)	The Ethernet MAC always returns a 1 for this bit. Together with bit 0.13, speed selection of 1000 Mb/s is identified.	Returns 1	1
0.5	Unidirectional Enable	Enable transmit regardless of whether a valid link has been established.	Read/Write	EMAC_UNIDIRECTION_ENABLE See Table 2-18, page 41.
0.4:0	Reserved	Always returns 0s, writes ignored.	Returns 0s	00000

Table 5-4: Status Register (Register 1)

Bit(s)	Name	Description	Type	Default Value
1.15	100BASE-T4	The Ethernet MAC always returns a 0 for this bit because 100BASE-T4 is not supported.	Returns 0	0
1.14	100BASE-X Full Duplex	The Ethernet MAC always returns a 0 for this bit because 100BASE-X full duplex is not supported.	Returns 0	0
1.13	100BASE-X Half Duplex	The Ethernet MAC always returns a 0 for this bit because 100BASE-X half duplex is not supported.	Returns 0	0
1.12	10 Mb/s Full Duplex	The Ethernet MAC always returns a 0 for this bit because 10 Mb/s full duplex is not supported.	Returns 0	0
1.11	10 Mb/s Half Duplex	The Ethernet MAC always returns a 0 for this bit because 10 Mb/s half duplex is not supported.	Returns 0	0
1.10	100BASE-T2 Full Duplex	The Ethernet MAC always returns a 0 for this bit because 100BASE-T2 full duplex is not supported.	Returns 0	0
1.9	100BASE-T2 Half Duplex	The Ethernet MAC always returns a 0 for this bit because 100BASE-T2 half duplex is not supported.	Returns 0	0
1.8	Extended Status	The Ethernet MAC always returns a 1 for this bit, indicating the presence of the extended register (Register 15).	Returns 1	1
1.7	Unidirectional Ability	Always returns a 1, writes ignored.	Returns 1	1
1.6	MF Preamble Suppression	The Ethernet MAC always returns a 1 for this bit to indicate the support of management frame preamble suppression.	Returns 1	1
1.5	Auto-Negotiation Complete	1 = Auto-negotiation process completed. 0 = Auto-negotiation process not completed.	Read Only	0



Table 5-4: Status Register (Register 1) (Cont'd)

Bit(s)	Name	Description	Type	Default Value
1.4	Remote Fault	1 = Remote fault condition detected. 0 = No remote fault condition detected.	Read Only Self Clearing on read	0
1.3	Auto-Negotiation Ability	The Ethernet MAC always returns a 1 for this bit, indicating that the PHY is capable of auto-negotiation.	Returns 1	1
1.2	Link Status	1 = PHY link is up. 0 = PHY link is down.	Read Only Self clearing on read	0
1.1	Jabber Detect	The Ethernet MAC always returns a 0 for this bit because jabber detect is not supported.	Returns 0	0
1.0	Extended Capability	The Ethernet MAC always returns a 0 for this bit because no extended register set is supported.	Returns 0	0

Table 5-5: PHY Identifier (Registers 2 and 3)

Bit(s)	Name	Description	Type	Default Value
2.15:0	Organizationally Unique Identifier	Organizationally Unique Identifier (OUI) from IEEE is 0x000A35.	Returns OUI(3-18)	0000000000101000
3.15:10	Organizationally Unique Identifier	Organizationally Unique Identifier (OUI) from IEEE is 0x000A35.	Returns OUI(19-24)	110101
3.9:4	Manufacturer's Model Number	Always returns 0s.	Returns 0s	000000
3.3:0	Revision Number	Always returns 0s.	Returns 0s	0000

Table 5-6: Auto-Negotiation Advertisement Register (Register 4)

Bit(s)	Name	Description	Type	Default Value
4.15	Next Page	1 = Next page functionality is advertised. 0 = Next page functionality is not advertised.	Read/Write	0
4.14	Reserved	Always returns 0, writes ignored.	Returns 0	0
4.13:12	Remote Fault	00 = No error. 01 = Offline. 10 = Link failure. 11 = Auto-negotiation error.	Read/Write Self clearing to 00 after auto-negotiation	00
4.11:9	Reserved	Always return 0, writes ignored.	Returns 0	0
4.8:7	Pause	00 = No PAUSE. 01 = Symmetric PAUSE. 10 = Asymmetric PAUSE towards link partner. 11 = Both symmetric PAUSE and asymmetric PAUSE towards link partner.	Read/Write	11
4.6	Half Duplex	The Ethernet MAC always returns a 0 for this bit because half-duplex mode is not supported.	Returns 0	0

Table 5-6: Auto-Negotiation Advertisement Register (Register 4) (Cont'd)

Bit(s)	Name	Description	Type	Default Value
4.5	Full Duplex	1 = Full-duplex mode is advertised. 0 = Full-duplex mode is not advertised.	Read/Write	1
4.4:0	Reserved	Always returns 0s, writes ignored.	Returns 0s	00000

Table 5-7: Auto-Negotiation Link Partner Ability Base Register (Register 5)

Bit(s)	Name	Description	Type	Default Value
5.15	Next Page	1 = Next page functionality is supported. 0 = Next page functionality is not supported.	Read Only	0
5.14	Acknowledge	Used by the auto-negotiation function to indicate reception of a link partner's base or next page.	Read Only	0
5.13:12	Remote Fault	00 = No error. 01 = Offline. 10 = Link failure. 11 = Auto-negotiation error.	Read Only	00
5.11:9	Reserved	Always returns 0s.	Returns 0s	000
5.8:7	Pause	00 = No PAUSE. 01 = Symmetric PAUSE supported. 10 = Asymmetric PAUSE supported. 11 = Both symmetric PAUSE and asymmetric PAUSE supported.	Read Only	00
5.6	Half Duplex	1 = Half-duplex mode is supported. 0 = Half-duplex mode is not supported.	Read Only	0
5.5	Full Duplex	1 = Full-duplex mode is supported. 0 = Full-duplex mode is not supported.	Read Only	0
5.4:0	Reserved	Always returns 0s.	Returns 0s	00000

Table 5-8: Auto-Negotiation Expansion Register (Register 6)

Bit(s)	Name	Description	Type	Default Value
6.15:3	Reserved	Always returns 0s.	Returns 0s	0000000000000
6.2	Next Page Able	The Ethernet MAC always returns a 1 for this bit because the device is Next Page Able.	Returns 1	1
6.1	Page Received	1 = A new page is received. 0 = A new page is not received.	Read only. Self clearing on read.	0
6.0	Reserved	Always returns 0s.	Returns 0s	0000000

Table 5-9: Auto-Negotiation Next Page Transmit Register (Register 7)

Bit(s)	Name	Description	Type	Default Value
7.15	Next Page	1 = Additional next page(s) will follow. 0 = Last page.	Read/Write	0
7.14	Reserved	Always returns 0.	Returns 0	0
7.13	Message Page	1 = Message Page. 0 = Unformatted Page.	Read/Write	1
7.12	Acknowledge 2	1 = Complies with message. 0 = Cannot comply with message.	Read/Write	0
7.11	Toggle	Value toggles between subsequent pages.	Read Only	0
7.10:0	Message or Unformatted Code Field	Message code field or unformatted page encoding as dictated by 7.13.	Read/Write	00000000001 (Null Message Code)

Table 5-10: Auto-Negotiation Next Page Receive Register (Register 8)

Bit(s)	Name	Description	Type	Default Value
8.15	Next Page	1 = Additional Next Page(s) will follow. 0 = Last page.	Read Only	0
8.14	Acknowledge	Used by auto-negotiation function to indicate reception of a link partner's base or next page.	Read Only	0
8.13	Message Page	1 = Message Page. 0 = Unformatted Page.	Read Only	0
8.12	Acknowledge 2	1 = Complies with message. 0 = Cannot comply with message.	Read Only	0
8.11	Toggle	Value toggles between subsequent next pages.	Read Only	0
8.10:0	Message / Unformatted Code Field	Message code field or unformatted page encoding as dictated by 8.13.	Read Only	00000000000

Table 5-11: Extended Status Register (Register 15)

Bit(s)	Name	Description	Type	Default Value
15.15	1000BASE-X Full Duplex	The Ethernet MAC always returns a 1 for this bit because 1000BASE-X full duplex is supported.	Returns 1	1
15.14	1000BASE-X Half Duplex	The Ethernet MAC always returns a 0 for this bit because 1000BASE-X half duplex is not supported.	Returns 0	0
15.13	1000BASE-T Full Duplex	The Ethernet MAC always returns a 0 for this bit because 1000BASE-T full duplex is not supported.	Returns 0	0
15.12	1000BASE-T Half Duplex	The Ethernet MAC always returns a 0 for this bit because 1000BASE-T half duplex is not supported.	Returns 0	0
15.11:0	Reserved	Always returns 0s.	Returns 0s	000000000000

Table 5-12: Vendor-Specific Register: Auto-Negotiation Interrupt Control Register (Register 16)

Bit(s)	Name	Description	Type	Default Value
16.15:2	Reserved	Always returns 0s.	Returns 0s	000000000000000
16.1	Interrupt Status	<p>1 = Interrupt is asserted. 0 = Interrupt is not asserted.</p> <p>If the interrupt is enabled, this bit is set upon the completion of an auto-negotiation cycle; it is cleared only by writing 0 to this bit.</p> <p>If the interrupt is disabled, this bit is reset to 0.</p> <p>The EMACCLIENTANINTERRUPT port is wired to this bit.</p>	Read/Write	0
16.0	Interrupt Enable	<p>1 = Interrupt is enabled. 0 = Interrupt is disabled.</p>	Read/Write	1

Table 5-13: Vendor-Specific Register: Loopback Control Register (Register 17)

Bit(s)	Name	Description	Type	Default Value
17.15:1	Reserved	Always returns 0s.	Returns 0s	000000000000000
17.0	Loopback Position	<p>0 = Loopback (when enabled) occurs in the Ethernet MAC directly before the interface to the serial transceiver.</p> <p>1 = Loopback (when enabled) occurs in the serial transceiver.</p> <p>Note: Loopback is enabled or disabled using 0.14 (see <a href="#">Control Register (Register 0)</a>).</p>	Read/Write	<p>EMAC_GTLOOPBACK</p> <p>See <a href="#">Table 2-18, page 41</a>.</p>

# SGMII Management Registers

Figure 5-8 shows the PCS/PMA sublayer block in an SGMII implementation. The PCS/PMA sublayer of the Ethernet MAC, when performing the SGMII standard, contains PCS Management registers as listed in Table 5-14. These registers are described in further detail in Table 5-15 through Table 5-25.

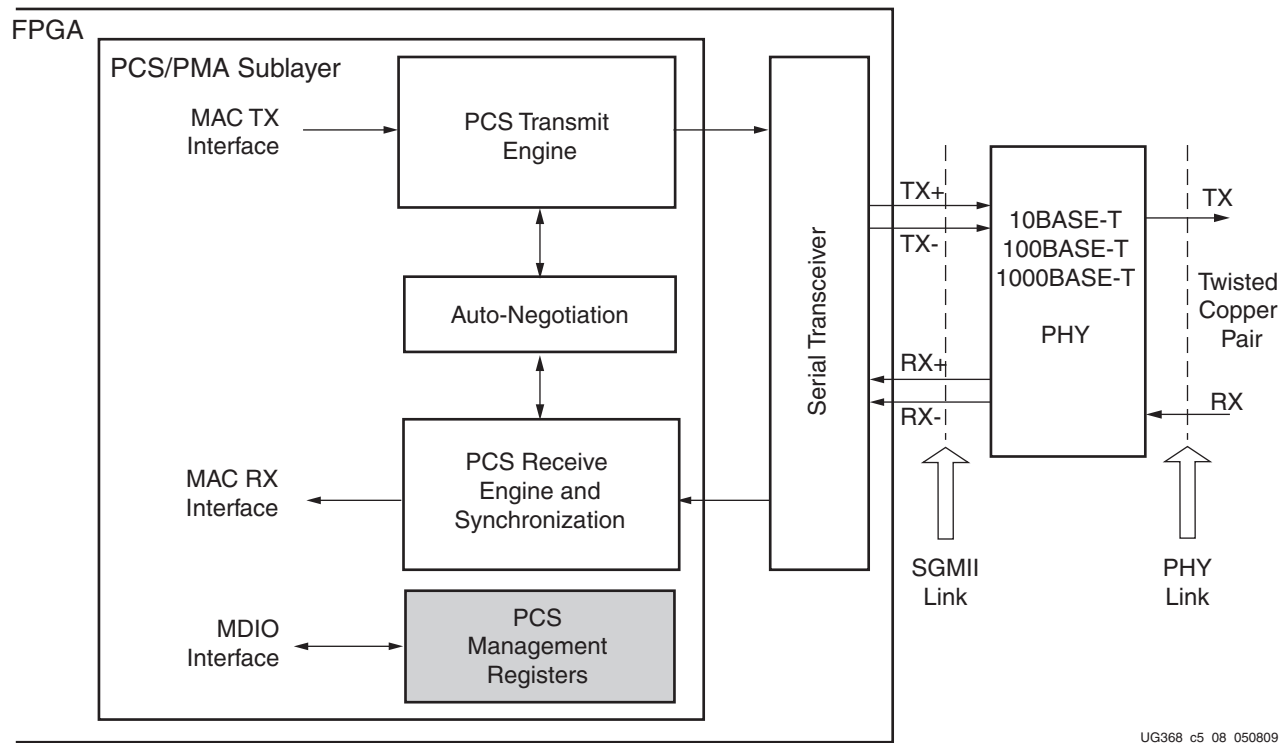


Figure 5-8: SGMII Logic and Physical Connections

These registers contain information relating to the operation of the system, including the status of both the SGMII link and the physical Ethernet link (PHY link). Refer to Figure 5-8.

Additionally, these registers are directly involved in the operation of the SGMII auto-negotiation function which occurs between the Ethernet MAC and the external PHY device, illustrated as a Tri-speed BASE-T PHY in Figure 5-8. (see SGMII Auto-Negotiation, page 182).

Table 5-14: SGMII Configuration Registers for 1000BASE-X PCS/PMA

Register Address (REGAD)	Register Name
0	SGMII Control Register (Register 0)
1	SGMII Status Register (Register 1)
2, 3	PHY Identifier (Registers 2 and 3)
4	SGMII Auto-Negotiation Advertisement Register (Register 4)
5	SGMII Auto-Negotiation Link Partner Ability Base Register (Register 5)

Table 5-14: SGMII Configuration Registers for 1000BASE-X PCS/PMA (Cont'd)

Register Address (REGAD)	Register Name
6	<a href="#">SGMII Auto-Negotiation Expansion Register (Register 6)</a>
7	<a href="#">SGMII Auto-Negotiation Next Page Transmit Register (Register 7)</a>
8	<a href="#">SGMII Auto-Negotiation Next Page Receive Register (Register 8)</a>
15	<a href="#">SGMII Extended Status Register (Register 15)</a>
16	<a href="#">SGMII Vendor-Specific Register: Auto-Negotiation Interrupt Control Register (Register 16)</a>
17	<a href="#">SGMII Vendor Specific Register: Loopback Control Register (Register 17)</a>

Table 5-15: SGMII Control Register (Register 0)

Bit(s)	Name	Description	Type	Default Value
0.15	Reset	1 = PCS/PMA reset. 0 = Normal operation.	Read/Write Self Clearing	EMAC_PHYRESET See <a href="#">Table 2-18, page 41</a> .
0.14	Loopback	1 = Enable loopback mode. 0 = Disable loopback mode.	Read/Write	EMAC_PHYLOOPBACKMSB See <a href="#">Table 2-18, page 41</a> .
0.13	Speed Selection (LSB)	The Ethernet MAC always returns a 0 for this bit. Along with bit 0.6, speed selection of 1000 Mb/s is identified.	Returns 0	0
0.12	Auto-Negotiation Enable	1 = Enable SGMII auto-negotiation process. 0 = Disable SGMII auto-negotiation process.	Read/Write	EMAC_PHYINITAUTONEG_ENABLE See <a href="#">Table 2-18, page 41</a> .
0.11	Power Down	1 = Power down. 0 = Normal operation. When set to 1, the serial transceiver is placed in a Low power state. This bit requires a reset (see bit 0.15) to clear.	Read/ Write	EMAC_PHYPOWERDOWN See <a href="#">Table 2-18, page 41</a> .
0.10	Isolate	1 = Isolate the SGMII logic from GMII. 0 = Normal operation.	Read/Write	EMAC_PHYISOLATE See <a href="#">Table 2-18, page 41</a> .
0.9	Restart Auto-Negotiation	1 = Restart Auto-negotiation process across the SGMII link. 0 = Normal operation.	Read/Write Self Clearing	0
0.8	Duplex Mode	The Ethernet MAC always returns a 1 for this bit to signal full-duplex mode.	Returns 1	1
0.7	Collision Test	The Ethernet MAC always returns a 0 for this bit to disable COL test.	Returns 0	0
0.6	Speed Selection (MSB)	The Ethernet MAC always returns a 1 for this bit. Together with bit 0.13, speed selection of 1000 Mb/s is identified.	Returns 1	1

Table 5-15: SGMII Control Register (Register 0) (Cont'd)

Bit(s)	Name	Description	Type	Default Value
0.5	Unidirectional Enable	Enable transmit regardless of whether a valid link has been established	Read/Write	EMAC_UNIDIRECTION_ENABLE See Table 2-18, page 41.
0.4:0	Reserved	Always returns 0s, writes ignored.	Returns 0s	00000

Table 5-16: SGMII Status Register (Register 1)

Bit(s)	Name	Description	Type	Default Value
1.15	100BASE-T4	The Ethernet MAC always returns a 0 for this bit because 100BASE-T4 is not supported.	Returns 0	0
1.14	100BASE-X Full Duplex	The Ethernet MAC always returns a 0 for this bit because 100BASE-X full duplex is not supported.	Returns 0	0
1.13	100BASE-X Half Duplex	The Ethernet MAC always returns a 0 for this bit because 100BASE-X half duplex is not supported.	Returns 0	0
1.12	10 Mb/s Full Duplex	The Ethernet MAC always returns a 0 for this bit because 10 Mb/s full duplex is not supported.	Returns 0	0
1.11	10 Mb/s Half Duplex	The Ethernet MAC always returns a 0 for this bit because 10 Mb/s half duplex is not supported.	Returns 0	0
1.10	100BASE-T2 Full Duplex	The Ethernet MAC always returns a 0 for this bit because 100BASE-T2 full duplex is not supported.	Returns 0	0
1.9	100BASE-T2 Half Duplex	The Ethernet MAC always returns a 0 for this bit because 100BASE-T2 half duplex is not supported.	Returns 0	0
1.8	Extended Status	The Ethernet MAC always returns a 1 for this bit, indicating the presence of the extended register (Register 15).	Returns 1	1
1.7	Unidirectional Ability	Always returns a 1, writes ignored.	Returns 1	1
1.6	MF Preamble Suppression	The Ethernet MAC always returns a 1 for this bit to indicate the support of management frame preamble suppression.	Returns 1	1
1.5	Auto-Negotiation Complete	1 = Auto-negotiation process completed. 0 = Auto-negotiation process not completed.	Read Only	0
1.4	Remote Fault	1 = Remote fault condition detected. 0 = No remote fault condition detected.	Read Only Self clearing on read	0
1.3	Auto-Negotiation Ability	The Ethernet MAC always returns a 1 for this bit, indicating that the PHY is capable of auto-negotiation.	Returns 1	1

Table 5-16: SGMII Status Register (Register 1) (Cont'd)

Bit(s)	Name	Description	Type	Default Value
1.2	SGMII Link Status	The link status of the Ethernet MAC with its external PHY across the SGMII link (see <a href="#">Figure 5-8</a> ). 1 = SGMII link is up. 0 = SGMII link is down.	Read Only Self clearing on read	0
1.1	Jabber Detect	The Ethernet MAC always returns a 0 for this bit because jabber detect is not supported.	Returns 0	0
1.0	Extended Capability	The Ethernet MAC always returns a 0 for this bit because no extended register set is supported.	Returns 0	0

Table 5-17: PHY Identifier (Registers 2 and 3)

Bit(s)	Name	Description	Type	Default Value
2.15:0	Organizationally Unique Identifier	Organizationally Unique Identifier (OUI) from IEEE is 0x000A35.	Returns OUI(3-18)	0000000000101000
3.15:10	Organizationally Unique Identifier	Organizationally Unique Identifier (OUI) from IEEE is 0x000A35.	Returns OUI(19-24)	110101
3.9:4	Manufacturer's Model Number	Always returns 0s.	Returns 0s	000000
3.3:0	Revision Number	Always returns 0s.	Returns 0s	0000

Table 5-18: SGMII Auto-Negotiation Advertisement Register (Register 4)

Bit(s)	Name	Description	Type	Default Value
4.15:0	All bits	SGMII defined value sent from the MAC to the PHY.	Read Only	0000000000000001

Table 5-19: SGMII Auto-Negotiation Link Partner Ability Base Register (Register 5)

Bit(s)	Name	Description	Type	Default Value
5.15	PHY Link Status	This refers to the link status of the external PHY device with its link partner across the PHY link (see <a href="#">Figure 5-8</a> ). 1 = Link up. 0 = Link down.	Read only	1
5.14	Acknowledge	Used by the auto-negotiation function to indicate reception of a link partner's base or next page.	Read only	0
5.13	Reserved	Always return 0.	Returns 0	0
5.12	Duplex mode	The resolved duplex mode that the external PHY device has auto-negotiated with its link partner across the PHY link (see <a href="#">Figure 5-8</a> ). 1 = Full Duplex. 0 = Half Duplex.	Read only	00



Table 5-19: SGMII Auto-Negotiation Link Partner Ability Base Register (Register 5) (Cont'd)

Bit(s)	Name	Description	Type	Default Value
5.11:10	Speed	The resolved operating speed that the external PHY device has auto-negotiated with its link partner across the PHY link (see Figure 5-8). 00 = 10 Mb/s. 01 = 100 Mb/s. 10 = 1000 Mb/s. 11 = Reserved.	Read only	00
5.9:1	Reserved	Always returns 0s.	Returns 0s	000000000
5.0	Reserved	Always returns 1.	Returns 1	1

Table 5-20: SGMII Auto-Negotiation Expansion Register (Register 6)

Bit(s)	Name	Description	Type	Default Value
6.15:3	Reserved	Always returns 0s.	Returns 0s	00000000000000
6.2	Next Page Able	The Ethernet MAC always returns a 1 for this bit because the device is Next Page Able.	Returns 1	1
6.1	Page Received	1 = A new page is received. 0 = A new page is not received.	Read only. Self clearing on read.	0
6.0	Reserved	Always returns 0s.	Returns 0s	0000000

Table 5-21: SGMII Auto-Negotiation Next Page Transmit Register (Register 7)

Bit(s)	Name	Description	Type	Default Value
7.15	Next Page	1 = Additional next page(s) will follow 0 = Last page.	Read/Write	0
7.14	Reserved	Always returns 0.	Returns 0	0
7.13	Message Page	1 = Message Page. 0 = Unformatted Page.	Read/Write	1
7.12	Acknowledge 2	1 = Complies with message. 0 = Cannot comply with message.	Read/Write	0
7.11	Toggle	Value toggles between subsequent pages.	Read Only	0
7.10:0	Message or Unformatted Code Field	Message code field or unformatted page encoding as dictated by 7.13.	Read/Write	00000000001 (Null Message Code)

Table 5-22: SGMII Auto-Negotiation Next Page Receive Register (Register 8)

Bit(s)	Name	Description	Type	Default Value
8.15	Next Page	1 = Additional Next Page(s) will follow. 0 = Last page.	Read Only	0
8.14	Acknowledge	Used by auto-negotiation function to indicate reception of a link partner's base or next page.	Read Only	0
8.13	Message Page	1 = Message Page. 0 = Unformatted Page.	Read Only	0

Table 5-22: SGMII Auto-Negotiation Next Page Receive Register (Register 8) (Cont'd)

Bit(s)	Name	Description	Type	Default Value
8.12	Acknowledge 2	1 = Complies with message. 0 = Cannot comply with message.	Read Only	0
8.11	Toggle	Value toggles between subsequent next pages.	Read Only	0
8.10:0	Message / Unformatted Code Field	Message code field or unformatted page encoding as dictated by 8.13.	Read Only	000000000000

Table 5-23: SGMII Extended Status Register (Register 15)

Bit(s)	Name	Description	Type	Default Value
15.15	1000BASE-X Full Duplex	The Ethernet MAC always returns a 1 for this bit because 1000BASE-X full duplex is supported.	Returns 1	1
15.14	1000BASE-X Half Duplex	The Ethernet MAC always returns a 0 for this bit because 1000BASE-X half duplex is not supported.	Returns 0	0
15.13	1000BASE-T Full Duplex	The Ethernet MAC always returns a 0 for this bit because 1000BASE-T full duplex is not supported.	Returns 0	0
15.12	1000BASE-T Half Duplex	The Ethernet MAC always returns a 0 for this bit because 1000BASE-T half duplex is not supported.	Returns 0	0
15.11:0	Reserved	Always returns 0s.	Returns 0s	000000000000

Table 5-24: SGMII Vendor-Specific Register: Auto-Negotiation Interrupt Control Register (Register 16)

Bit(s)	Name	Description	Type	Default Value
16.15:2	Reserved	Always returns 0s.	Returns 0s	00000000000000
16.1	Interrupt Status	1 = Interrupt is asserted. 0 = Interrupt is not asserted.  If the interrupt is enabled, this bit is set upon the completion of an auto-negotiation cycle. Writing 0 to this bit is the only way to clear it.  If the interrupt is disabled, this bit is cleared to 0.  The EMACCLIENTANINTERRUPT port is wired to this bit.	Read/Write	0
16.0	Interrupt Enable	1 = Interrupt is enabled. 0 = Interrupt is disabled.	Read/Write	1

Table 5-25: SGMII Vendor Specific Register: Loopback Control Register (Register 17)

Bit(s)	Name	Description	Type	Default Value
17.15:1	Reserved	Always returns 0s.	Returns 0s	0000000000000000
17.0	Loopback Position	<p>0 = Loopback (when enabled) occurs in the Ethernet MAC directly before the interface to the serial transceiver.</p> <p>1 = Loopback (when enabled) occurs in the serial transceiver.</p> <p>Note: Loopback is enabled or disabled using 0.14 (see <a href="#">Control Register (Register 0)</a>).</p>	Read/Write	<p>EMAC_GTLOOPBACK</p> <p>For physical interface attributes, see <a href="#">Table 2-18, page 41</a>.</p>



## Physical Interface

---

This chapter describes the physical interface options provided by the Ethernet MAC. This chapter contains the following sections:

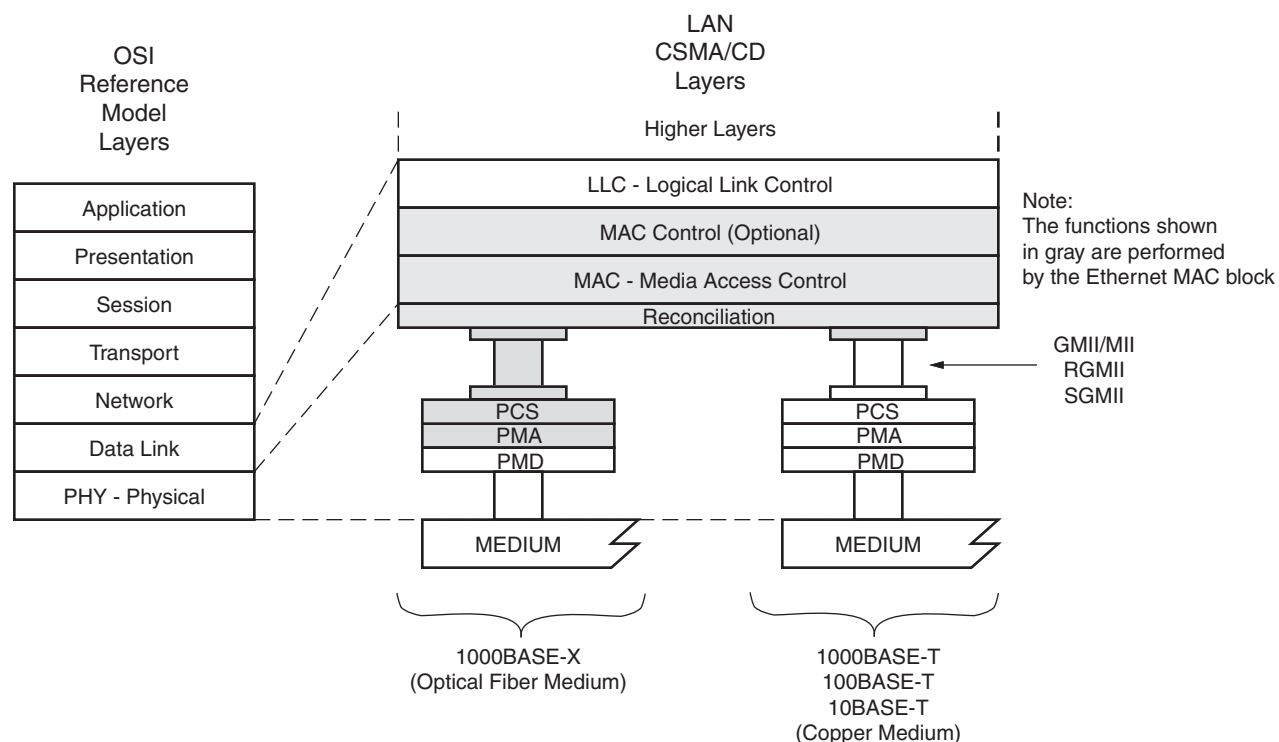
- [Introduction to the Physical Interfaces](#)
- [Media Independent Interface \(MII\)](#)
- [Gigabit Media Independent Interface \(GMII\)](#)
- [Reduced Gigabit Media Independent Interface \(RGMII\)](#)
- [1000BASE-X PCS/PMA](#)
- [Serial Gigabit Media Independent Interface \(SGMII\)](#)

For each physical interface, there can be more than one possible clocking scheme. The following sections describe the different optimized clocking schemes for the individual Ethernet MAC configurations.

Ethernet MAC wrappers and example designs are provided for each of the different physical interfaces via the CORE Generator™ tool. The generated designs exactly match the descriptions in this chapter and are available in both VHDL and Verilog. By using the CORE Generator tool, the time required to instantiate the Ethernet MAC into a usable design is greatly reduced. See [Accessing the Ethernet MAC from the CORE Generator Tool](#), page 21.

## Introduction to the Physical Interfaces

Figure 6-1 illustrates the position of the Ethernet MAC sublayers and physical interface options within the OSI model.



UG368\_c6\_01\_051209

Figure 6-1: MAC Sublayer Partitioning, Relationship to the ISO/IEC OSI Reference Model

The Ethernet MAC sublayer itself is independent of, and can connect to, any type of physical layer device. In Figure 6-1, two alternative physical sublayer standards are shown:

- **BASE-T** PHYs provide a link between the MAC and copper mediums. This functionality is not offered within the Ethernet MAC. However, external BASE-T PHY devices are readily available on the market. These can connect to the Ethernet MAC using GMII/MII, RGMII, or SGMII interfaces.
- **BASE-X** PHYs provide a link between the MAC and (usually) fiber optic mediums. The Ethernet MAC itself can support the 1 Gb/s BASE-X standard. 1000BASE-X can be offered by connecting the Ethernet MAC to a serial transceiver.

The interface type and 1000BASE-X sublayer functionalities are summarized in the following subsections.

### GMII / MII

The Media Independent Interface (MII), defined in IEEE Std 802.3-2005, clause 22 is a parallel interface that connects a 10-Mb/s and/or 100-Mb/s capable MAC to the physical sublayers.

The Gigabit Media Independent Interface (GMII), defined in IEEE Std 802.3-2005, clause 35 is an extension of the MII used to connect a 1-Gb/s capable MAC to the physical sublayers.

MII can be considered a subset of GMII, and as a result, GMII/MII together can carry Ethernet traffic at 10 Mb/s, 100 Mb/s, and 1 Gb/s.

## RGMII

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

RGMII can carry Ethernet traffic at 10 Mb/s, 100 Mb/s, and 1 Gb/s.

## SGMII

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

SGMII can carry Ethernet traffic at 10 Mb/s, 100 Mb/s, and 1 Gb/s.

## 1000BASE-X Sublayers

### PCS/PMA

The Physical Coding Sublayer (PCS) for 1000BASE-X operation is defined in IEEE Std 802.3-2005, clauses 36 and 37, and performs the following:

- Encoding (and decoding) of GMII data octets to form a sequence of ordered sets
- 8B/10B encoding (and decoding) of the sequence ordered sets
- 1000BASE-X auto-negotiation for information exchange with the link partner

The Physical Medium Attachment (PMA) for 1000BASE-X operation is defined in IEEE Std 802.3-2005, clause 36 and performs the following:

- Serialization (and deserialization) of code-groups for transmission (and reception) on the underlying serial PMD
- Recovery of clock from the 8B/10B-coded data supplied by the PMD

1000BASE-X PCS/PMA functionality is provided by connecting the Ethernet MAC to a serial transceiver.

### PMD

The Physical Medium Dependent (PMD) sublayer is defined in IEEE Std 802.3-2005, clause 38 for 1000BASE-LX and 1000BASE-SX (long and short wavelength laser). This type of PMD is provided by the external GBIC or SFP optical transceiver, which should be connected directly to the ports of the serial transceiver.

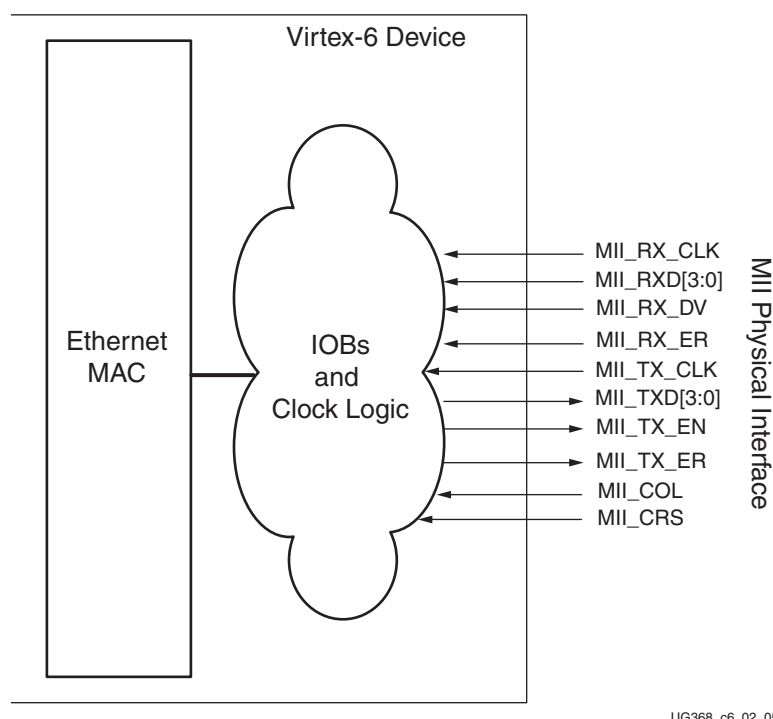
## Media Independent Interface (MII)

The Media Independent Interface (MII), defined in IEEE Std 802.3-2005, clause 22, is a parallel interface that connects a 10-Mb/s and/or 100-Mb/s capable MAC to the physical sublayers.

The Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC supports the MII physical interface at 2.5V only. Care must be taken to select a PHY that is electrically compatible

with the Virtex-6 FPGA LVCMOS25 voltage standard. See the *Virtex-6 FPGA Data Sheet: DC and Switching Characteristics* for more information.

Figure 6-2 shows the Ethernet MAC configured for MII. The physical signals of the Ethernet MAC are connected through IOBs to the external interface. The “logic cloud” indicates that alternative implementations are possible.



UG368\_c6\_02\_050809

Figure 6-2: Ethernet MAC Configured in MII Mode

There are two alternative implementations available that use different clocking schemes: the standard clocking scheme without clock enables or an alternative clock enable scheme that saves on global clock resources. [Appendix B, Ethernet MAC Clocks](#), introduces these two clocking models, which are described in detail in the following sections:

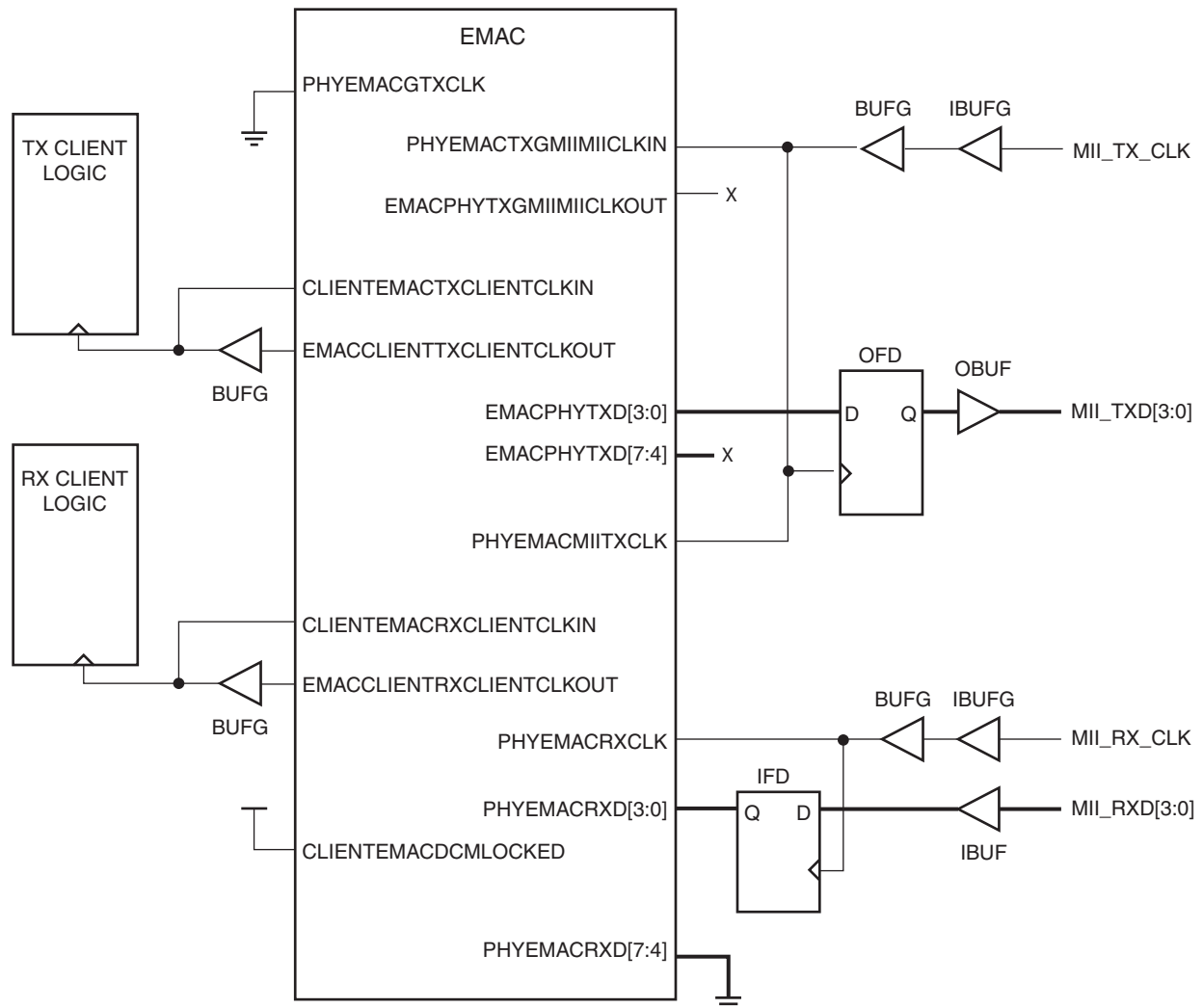
- [MII Standard Clock Management](#)
- [MII Clock Management Using Clock Enables](#)

If the CORE Generator tool is used, the wrapper files for the Ethernet MAC that are created will contain the logic described in these sections. By using the CORE Generator tool, the time required to instantiate the Ethernet MAC into a usable design is greatly reduced. See [Accessing the Ethernet MAC from the CORE Generator Tool](#), page 21.



## MII Standard Clock Management

Figure 6-3 shows the standard clock management scheme when the MII interface is selected. Both the MII\_TX\_CLK and MII\_RX\_CLK, generated from the PHY, have a frequency of either 2.5 MHz or 25 MHz, depending on the operating speed of the Ethernet MAC. MII\_TX\_CLK connects to PHYEMACTXGMIIICLKIN and the MII\_TXD registers through a BUFG. MII\_RX\_CLK connects to PHYEMACRXCLK and the MII\_RXD registers through a BUFG. A regional buffer (BUFR) can replace the BUFG in either or both of these cases. Refer to the Virtex-6 FPGA User Guide for BUFR usage guidelines. EMACCLIENTTXCLIENTCLKOUT connects to EMACCLIENTTXCLIENTCLKIN through a BUFG. Its frequency is either 12.5 MHz or 1.25 MHz, depending on the operating speed of the Ethernet MAC. The RX client clocking is similar. The CLIENTEMACDCMLOCKED port must be tied High.

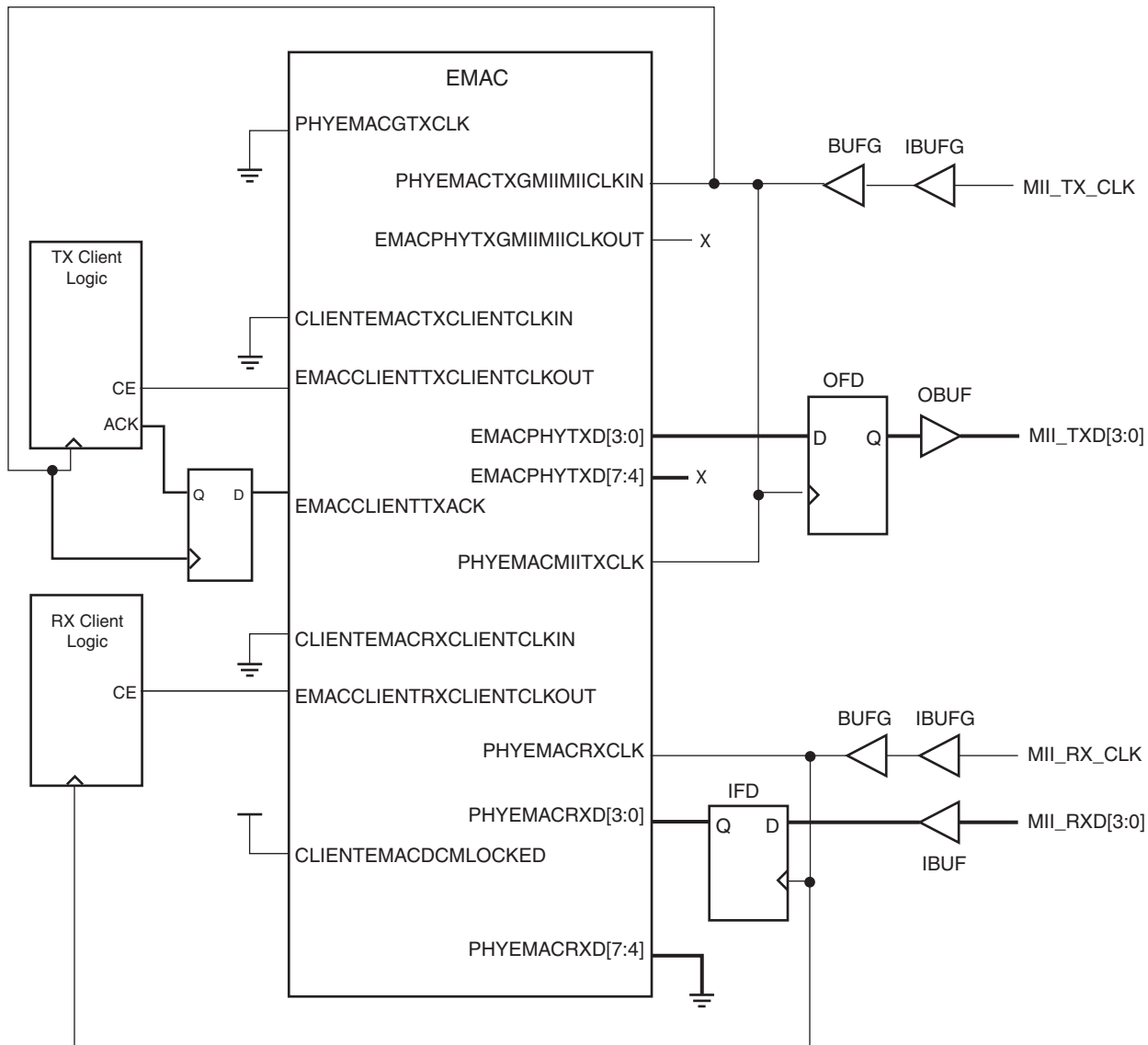


UG368\_c6\_03\_051309

Figure 6-3: MII Clock Management

## MII Clock Management Using Clock Enables

It is possible to use only two BUFGs by setting the EMAC\_USECLKEN attribute; however, the client and MII logic must be constrained to run at 125 MHz. Figure 6-4 shows the MII clock management scheme when operating in this mode. A regional buffer (BUFR) can replace the BUFG in either or both of these cases. Refer to the Virtex-6 FPGA User Guide for BUFR usage guidelines.



UG368\_c6\_04\_051309

Figure 6-4: MII Clock Management with Clock Enable

Using the example in Figure 6-4, all logic is clocked on the MII interface clocks. The frequencies of these clocks are 25 MHz at 100 Mb/s and 2.5 MHz at 10 Mb/s, or twice as fast as the client clock requires. To produce the correct clock frequency at the client, the client logic must be clock enabled to achieve the correct data rate. The clock enables for the client logic are provided by the EMACCLIENTTXCLIENTCLKOUT and EMACCLIENTRXCLIENTCLKOUT outputs.

Due to the timing relationship between the MII transmit clock and the internal client clock signals, the TX\_ACK signal (EMACCLIENTTXACK) is registered at the output of the Ethernet MAC.

## Gigabit Media Independent Interface (GMII)

The Media Independent Interface (MII), defined in IEEE Std 802.3-2005, clause 22, is a parallel interface that connects a 10-Mb/s and/or 100-Mb/s capable MAC to the physical sublayers. The Gigabit Media Independent Interface (GMII), defined in IEEE Std 802.3-2005, clause 35, is an extension of the MII used to connect a 1-Gb/s 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 Mb/s, 100 Mb/s, and 1 Gb/s.

The Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC supports the GMII/MII physical interface at 2.5V only. Care must be taken to select a PHY that is electrically compatible with the Virtex-6 FPGA LVCMOS25 voltage standard. See the *Virtex-6 FPGA Data Sheet: DC and Switching Characteristics* for more information.

Figure 6-5 shows the Ethernet MAC configured for GMII. The physical signals of the Ethernet MAC are connected through IOBs to the external interface. The “logic cloud” indicates that alternative implementations are possible.

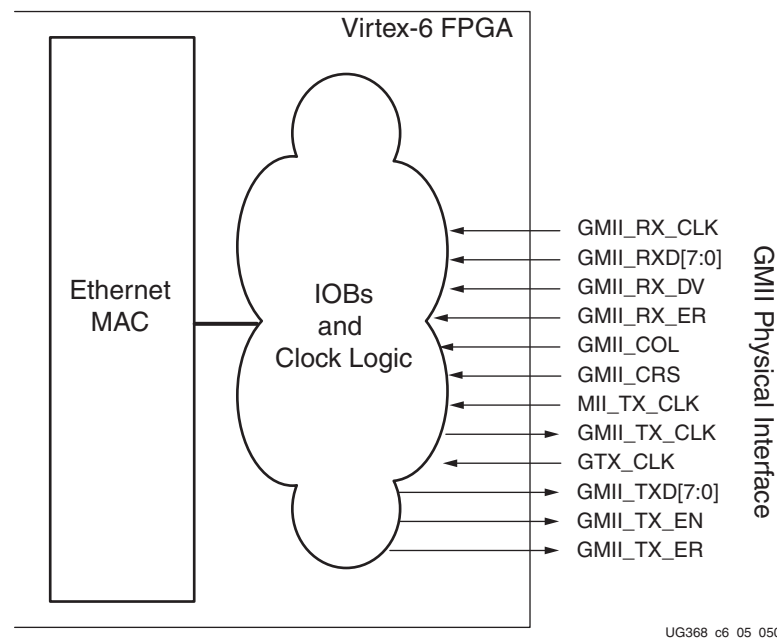


Figure 6-5: Ethernet MAC Configured in GMII Mode

There are several implementations available using different clocking schemes.

[Appendix B, Ethernet MAC Clocks](#), introduces these clocking models. They are described in detail in the following sections:

- [GMII Clock Management for 1 Gb/s Only](#)

At 1 Gb/s speeds only, client interface and physical interface clocks always run at the same frequency (125 MHz). This enables efficient clock logic implementations.

- [GMII Standard Clock Management for Tri-Speed Operation](#)

The standard clocking scheme for tri-speed operation.

- [GMII Clock Management for Tri-Speed Operation Using Clock Enables](#)

An alternative advanced clocking scheme for tri-speed operation, which saves on global clock resources.

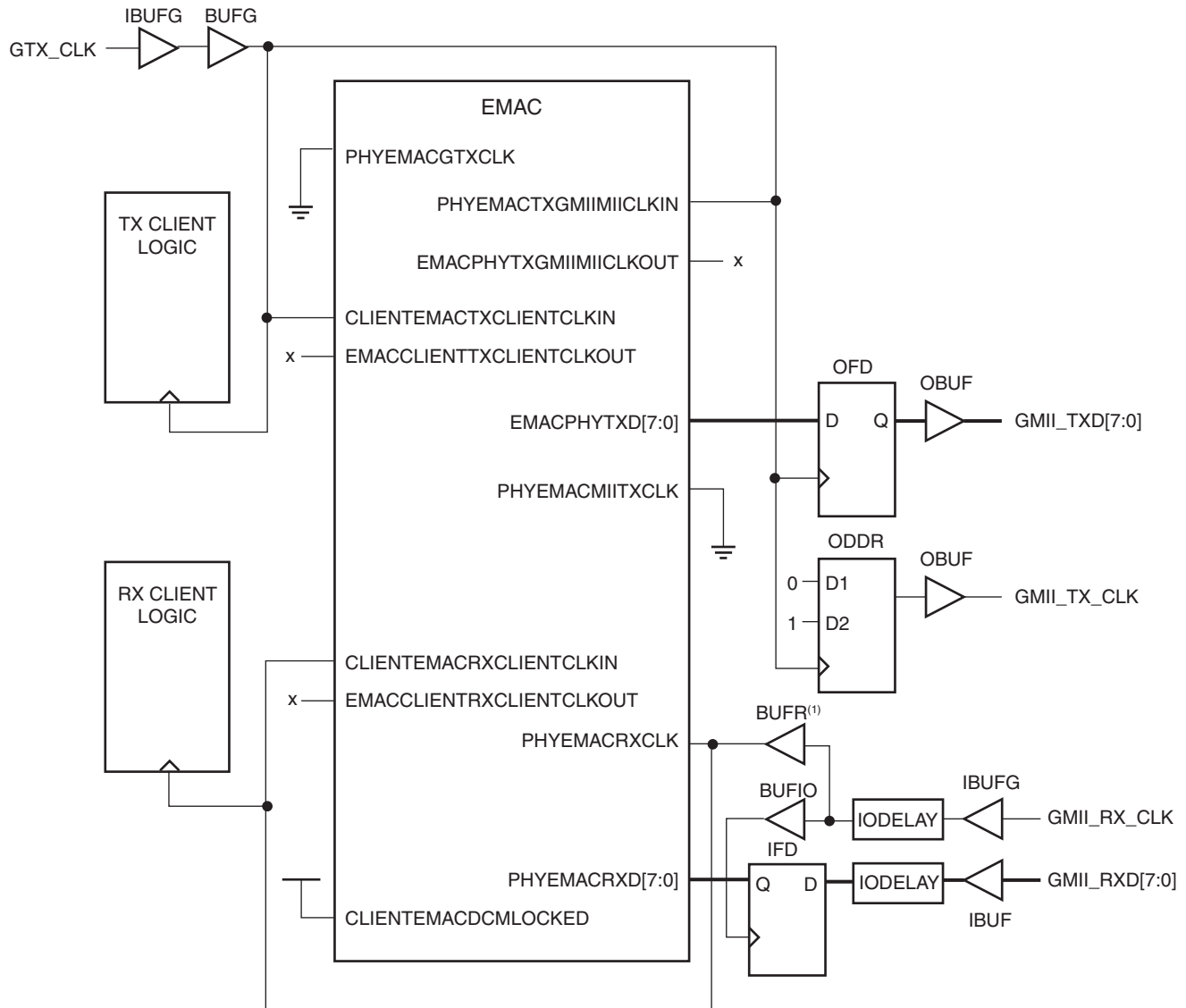
Regional clocking is used to implement the GMII physical interface. It is important to understand the pinout requirements of regional clocking models. For more information, see [Appendix A, Pinout Requirements](#).

If the CORE Generator tool is used, the wrapper files for the Ethernet MAC that are created will contain the logic described in these sections. By using the CORE Generator tool, the time required to instantiate the Ethernet MAC into a usable design is greatly reduced. See [Accessing the Ethernet MAC from the CORE Generator Tool, page 21](#).

## GMII Clock Management for 1 Gb/s Only

[Figure 6-6](#) shows GMII clock management when using a single Ethernet MAC.

When using this clocking methodology, which routes the GMII\_RX\_CLK net through a BUFIO and a BUFR, regional clocking requirements and their impact on pinout planning must be understood. For more information, see [Appendix A, Pinout Requirements](#).



Note 1: Pinout and clock region planning is required to accommodate regional clocking. See Appendix A, "Pinout Requirements," for details.

UG368\_c6\_06\_050809

Figure 6-6: 1 Gb/s GMII Clock Management

GTX\_CLK must be provided to the Ethernet MAC. This high-quality, 125 MHz clock satisfies the IEEE Std 802.3-2005 requirements.

The GTX\_CLK input is routed onto the global clock network via an IBUFG and BUFG. The output of the BUFG connects to:

- GMII\_TXD registers in the FPGA logic
- PHYEMACTXGMIIMICLKIN input port
- CLIENTEMACTXCLIENTCLKIN
- TX client logic

GMII\_TX\_CLK is forwarded along with the GMII data signals from the FPGA to the PHY. An IOB DDR output register is used; this is a predictable way to produce the clock because

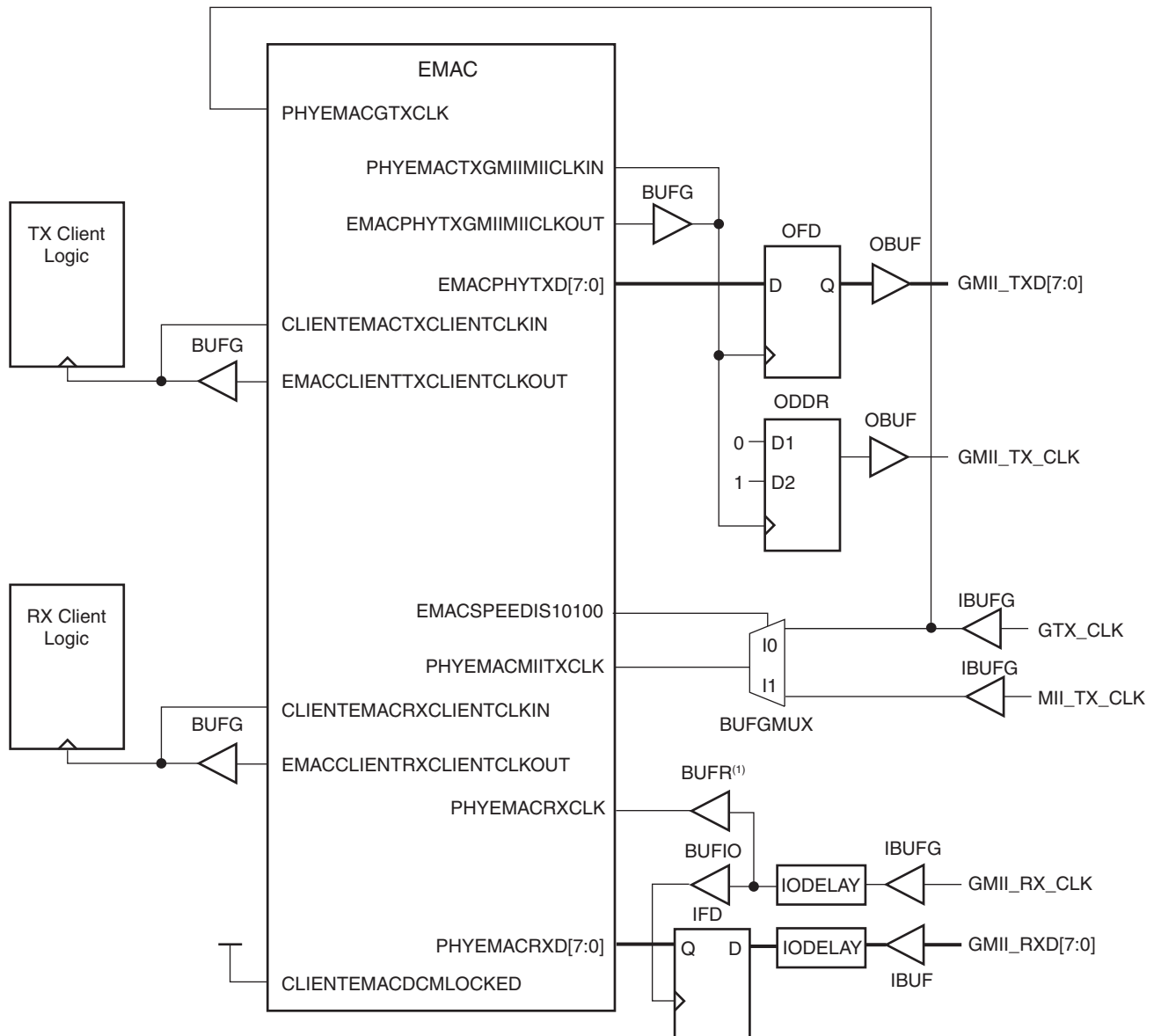
the clock-to-pad delay is the same as that for the GMII\_TXD signals. This forwarded clock is inverted with respect to PHYEMACTXGMIIIMCLKIN so that its rising edge occurs in the center of the GMII\_TXD data valid window. This produces the best possible setup and hold times for driving the external interface.

PHYEMACMIITXCLK is unused and must be tied to ground. The GMII\_RX\_CLK is generated from the PHY and connected to PHYEMACRXCLK through an IBUFG and BUFR. This regional clock net is also used to clock FPGA receiver logic, including the receiver client. A BUFIO is used to register the GMII\_RXD inputs, and fixed-mode IODELAYs are used on the GMII\_RX\_CLK and GMII\_RXD inputs to align the clock and data. These are set to sample a 2 ns setup, 0 ns hold window at the device pads. The CLIENTEMACDCMLOCKED port must be tied High.

## GMII Standard Clock Management for Tri-Speed Operation

[Figure 6-7](#) shows the clock management used with the tri-speed GMII interface. GTX\_CLK must be provided to the Ethernet MAC with a high-quality, 125 MHz clock that satisfies the IEEE Std 802.3-2005 requirements.

When using this clocking methodology, which routes the GMII\_RX\_CLK net through a BUFIO and a BUFR, regional clocking requirements and their impact on pinout planning must be understood. For more information, please see [Appendix A, Pinout Requirements](#).



Note 1: Pinout and clock region planning is required to accommodate regional clocking. See Appendix A, "Pinout Requirements," for details.

UG368\_c6\_07\_051109

Figure 6-7: GMII Tri-Mode Operation Clock Management

The EMACPHYTXGMIIMICLKOUT output port connects to the GMII logic in the FPGA logic and the PHYEMACTXGMIIMICLKIN input port through a BUFG. A regional buffer (BUFR) can replace this BUFG. Refer to the Virtex-6 FPGA User Guide for BUFR usage guidelines.

The EMACCLIENTTXCLIENTCLKOUT output port connects to the CLIENTEMACTXCLIENTCLKIN input port and transmit client logic in the FPGA logic through a BUFG. The receive client clocking is similar.

The MII\_TX\_CLK signal generated from the PHY has a frequency of either 2.5 MHz or 25 MHz, depending on the operating speed of the Ethernet MAC. PHYEMACMIITXCLK is driven by a BUFGMUX that selects MII\_TX\_CLK during 10 or 100 Mb/s operation, or

GTX\_CLK during 1000 Mb/s operation. The GTX\_CLK always drives PHYEMACGTXCLK. The frequency of GMII\_RX\_CLK, generated from the PHY, is 2.5 MHz, 25 MHz, or 125 MHz, depending on the operating speed of the Ethernet MAC. Fixed-mode IODELAYS are used on the GMII\_RX\_CLK and GMII\_RXD inputs to align the clock and data. These are set to sample a 2 ns setup, 0 ns hold window at the device pads. The CLIENTEMACDCMLOCKED port must be tied High.

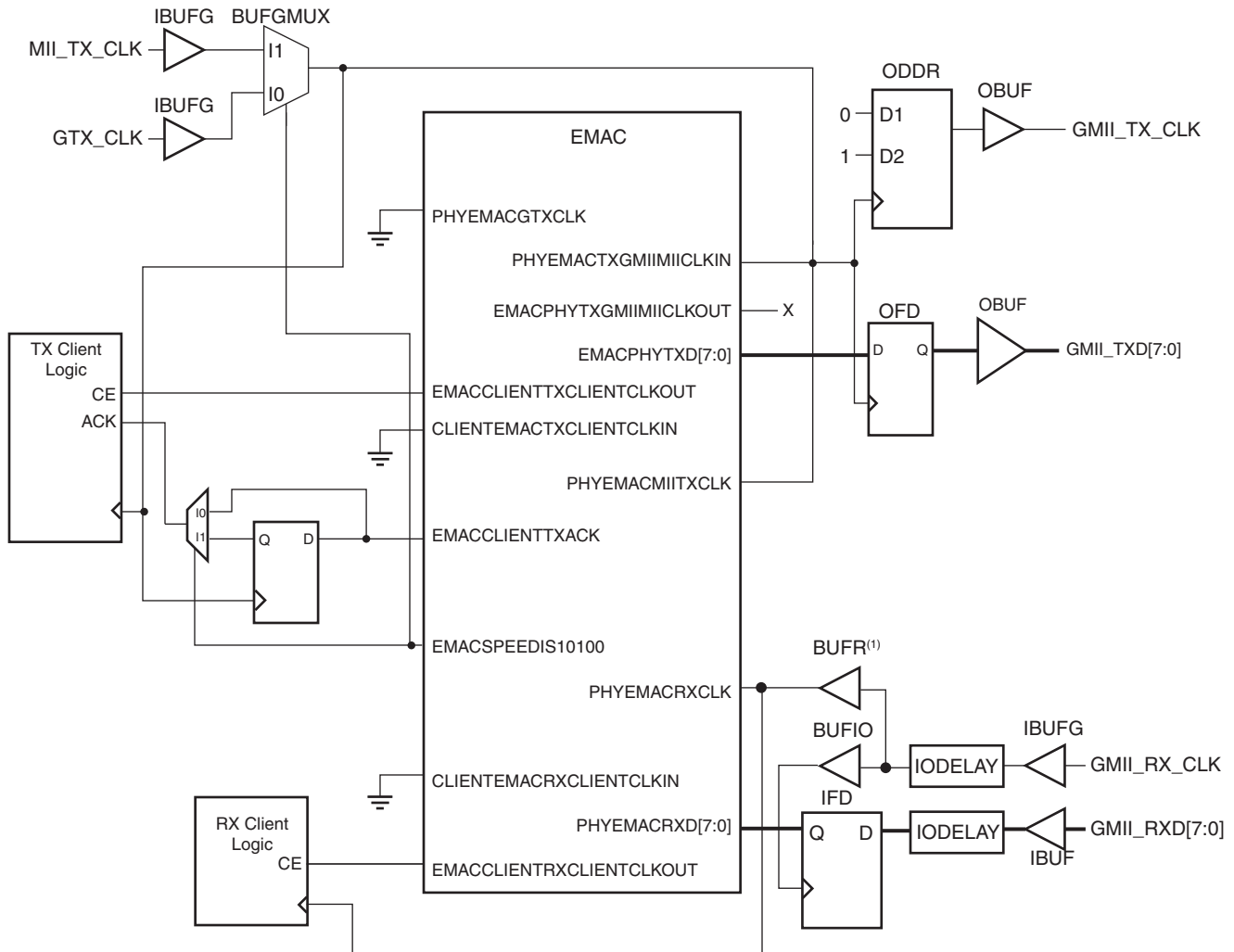
The GMII\_TX\_CLK is derived from the Ethernet MAC, routed through an OBUF, and then connected to the PHY. Because GMII\_TX\_CLK is derived from EMACPHYTXGMIIICLKOUT, its frequency automatically changes between 125 MHz, 25 MHz, or 2.5 MHz, depending on the speed setting of the Ethernet MAC.

## GMII Clock Management for Tri-Speed Operation Using Clock Enables

[Figure 6-8](#) shows the clock management scheme for tri-speed operation when the EMAC\_USECLKEN attribute is set. In this mode of operation, the transmitter signals are synchronous to PHYEMACTXGMIIICLKIN and the receiver signals are synchronous to PHYEMACRXCLK.

When using this clocking methodology, which routes the GMII\_RX\_CLK net through a BUFIO and a BUFR, regional clocking requirements and their impact on pinout planning must be understood. For more information, please see [Appendix A, Pinout Requirements](#).





Note 1: Pinout and clock region planning is required to accommodate regional clocking. See Appendix A, "Pinout Requirements," for details.

UG368\_c6\_08\_050809

Figure 6-8: Tri-Mode GMII Clock Management with Clock Enable

At 1 Gb/s, all external logic is clocked at 125 MHz. At 100 Mb/s and 10 Mb/s, the logic is clocked at 25 MHz and 2.5 MHz, respectively. To maintain the correct data rate at the client, interface clock enable signals are output on EMACCLIENTTXCLIENTCLKOUT for the transmitter logic and on EMACCLIENTRXCLIENTCLKOUT for the receiver logic. These signals are High during 1 Gb/s operation and toggle on each clock edge at slower speeds. These are used to enable the client interface logic.

Due to the timing relationship between the GMII transmit clock and the internal client clock signals, the TX\_ACK signal (EMACCLIENTTXACK) is registered at the output of the Ethernet MAC when operating at speeds below 1 Gb/s. At 1 Gb/s, the register is bypassed.

The GMII\_TX\_CLK is derived from the GTX\_CLK input when the MAC is operating at 1 Gb/s and from the MII\_TX\_CLK input when the MAC is operating at a lower speed. It is routed through an OBUF and then connected to the PHY. Its frequency changes between 125 MHz, 25 MHz, or 2.5 MHz depending on the speed setting of the Ethernet MAC.

This clocking scheme requires two fewer BUFGs than the default scheme detailed in [Figure 6-7](#).

## Reduced Gigabit Media Independent Interface (RGMII)

Reduced GMII (RGMII), defined by Hewlett-Packard, is an alternative to GMII. It reduces the number of pins required to connect the Ethernet MAC to the PHY from 24 to 12, when compared with GMII. RGMII achieves this 50% pin count reduction in the interface by using DDR flip-flops. For this reason, RGMII is often preferred over GMII by PCB designers. RGMII can carry Ethernet traffic at 10 Mb/s, 100 Mb/s, and 1 Gb/s. For more information on RGMII, refer to the *Hewlett Packard RGMII Specification, version 1.3 and 2.0*.

It is important to select the proper Virtex-6 FPGA voltage standard when using the RGMII physical interface. RGMII v1.3 is electrically compatible with the LVCMOS25 voltage standard, while RGMII v2.0 is electrically compatible with the HSTL\_I voltage standard. When using the CORE Generator tool, the provided UCF file includes the necessary constraints.

[Figure 6-9](#) shows the Ethernet MAC configured for RGMII. The physical signals of the Ethernet MAC are connected through IOBs to the external interface. The “logic cloud” indicates that alternative implementations are possible.

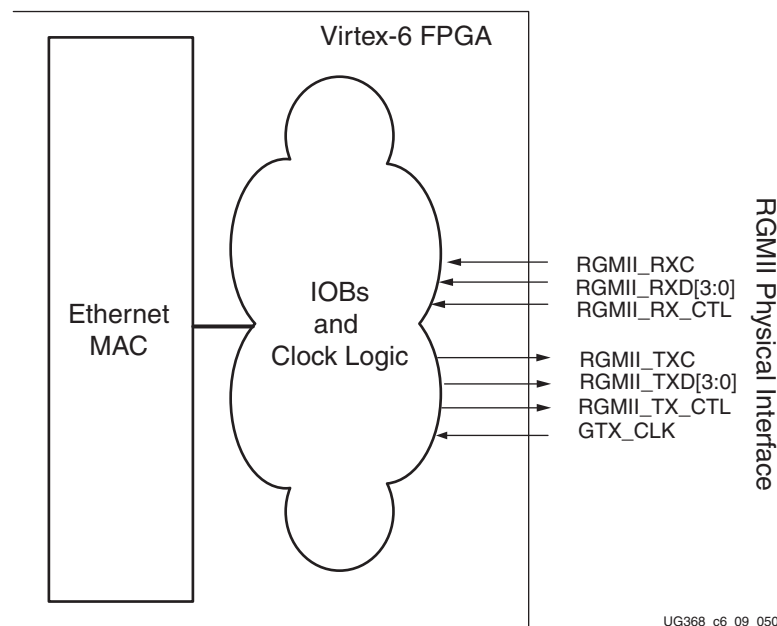


Figure 6-9: Ethernet MAC Configured in RGMII Mode

There are several implementations for both RGMII v1.3 and v2.0, which use alternative clocking schemes. [Appendix B, Ethernet MAC Clocks](#), introduces these clocking models. They are described in detail in the following sections:

- [RGMII Clock Management for 1 Gb/s Only](#)

At 1 Gb/s speeds only, client interface and physical interface clocks always run at the same frequency (125 MHz). This enables efficient clock logic implementations.

- [RGMII Standard Clock Management for Tri-Speed Operation](#)

The standard clocking scheme for tri-speed operation.

- [RGMII Clock Management for Tri-Speed Operation Using Clock Enables](#)

An advanced clocking scheme for tri-speed operation that saves global clock resources.

Regional clocking is used to implement the RGMII physical interface. Consequently, it is important to understand the pinout requirements of regional clocking models. For more information, see [Appendix A, Pinout Requirements](#).

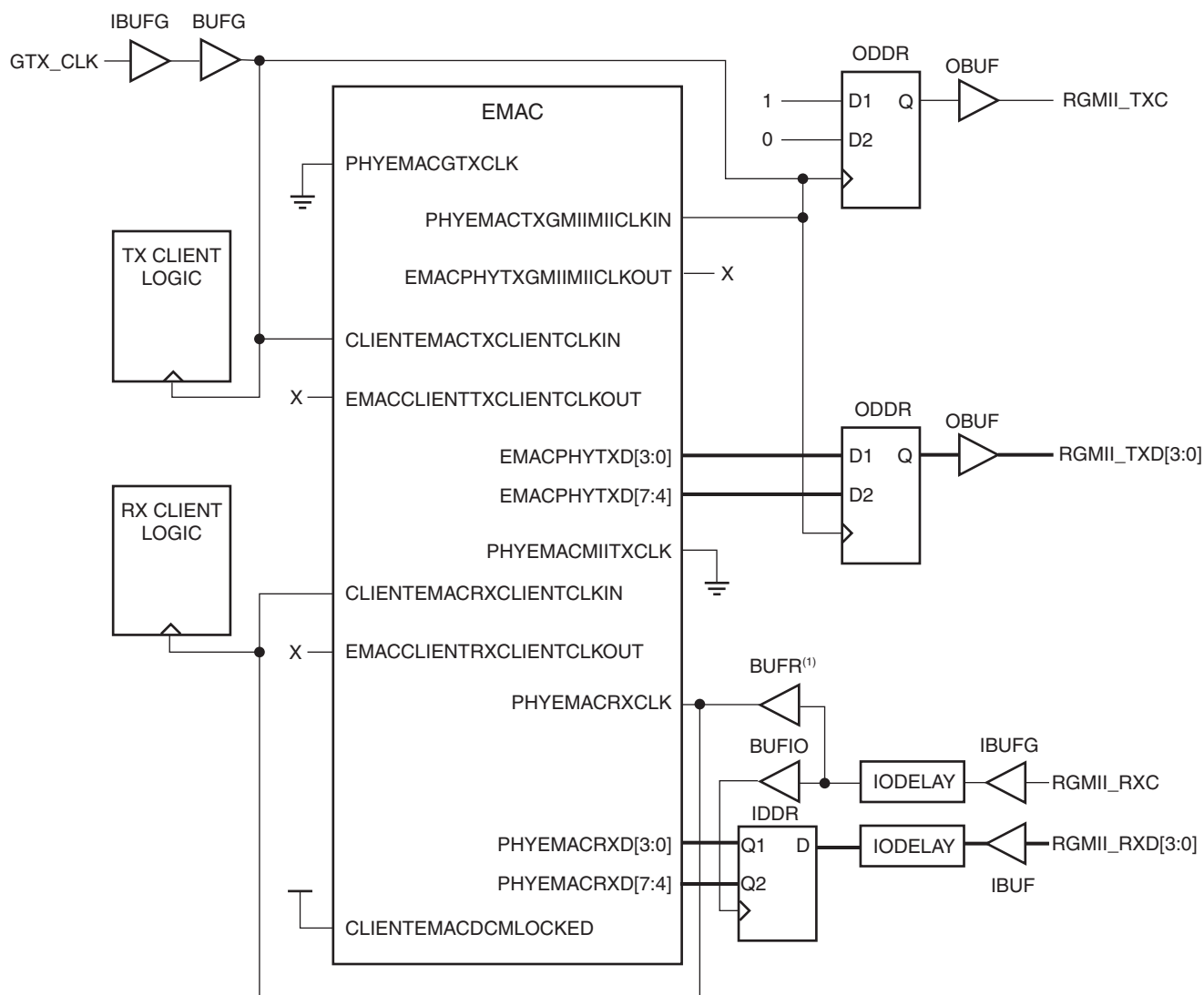
If the CORE Generator tool is used, then the wrapper files for the Ethernet MAC that are created contain the logic described in these sections. By using the CORE Generator tool, the time required to instantiate the Ethernet MAC into a usable design is greatly reduced. See [Accessing the Ethernet MAC from the CORE Generator Tool](#), page 21.

## RGMII Clock Management for 1 Gb/s Only

### RGMII Version 1.3

[Figure 6-10](#) shows the clock management used with the RGMII interface when using the *Hewlett Packard RGMII Specification v1.3*. GTX\_CLK must be provided to the Ethernet MAC with a high-quality, 125 MHz clock that satisfies the IEEE Std 802.3-2005 requirements. It is routed through an IBUFG and a BUFG and drives all transmitter logic.

When using this clocking methodology, which routes the RGMII\_RXC net through a BUFIO and a BUFR, it is important to understand the regional clocking requirements and their impact on pinout planning. For more information, see [Appendix A, Pinout Requirements](#).



Note 1: Pinout and clock region planning is required to accommodate regional clocking. See Appendix A, "Pinout Requirements," for details.

UG368\_c6\_10\_050809

**Figure 6-10: 1 Gb/s RGMII Hewlett Packard v1.3 Clock Management**

RGMII\_TXC is derived from GTX\_CLK by routing to an IOB DDR output register followed by an OBUF (which is then connected to the PHY). The use of the DDR register ensures that the forwarded clock is exactly in line with the RGMII transmitter data, as specified in the *Hewlett Packard RGMII Specification, v1.3*.

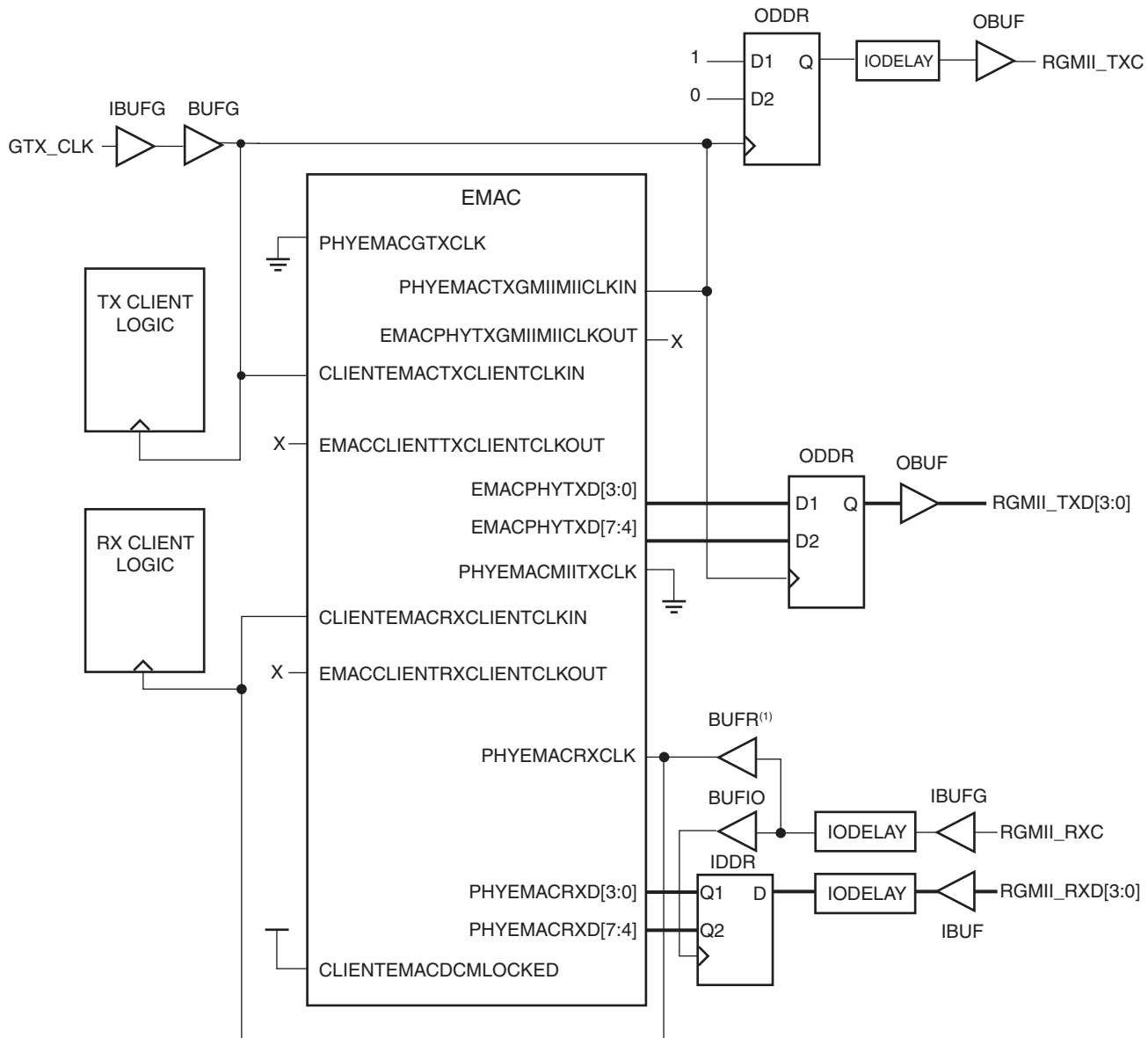
RGMII\_RXC is generated by the PHY and connected to PHYEMACRXCLK via an IBUFG and BUFR. This regional clock net is also used to clock FPGA receiver logic, including the receiver client. A BUFIO is used to register the RGMII\_RXD inputs, and fixed-mode IODELAYs are instantiated on the RGMII clock and data lines. These are set to sample a 1 ns setup, 1 ns hold window at the device pads.

The CLIENTEMACDCMLOCKED port must be tied High.

## RGMII Version 2.0

Figure 6-11 shows the clock management used with the RGMII interface when following the *Hewlett Packard RGMII Specification v2.0*. GTX\_CLK must be provided to the Ethernet MAC with a high-quality, 125 MHz clock that satisfies the IEEE Std 802.3-2005 requirements. It is routed through an IBUFG and a BUFG, and drives all transmitter logic.

When using this clocking methodology, which routes the RGMII\_RXC net through a BUFIO and a BUFR, it is important to understand the regional clocking requirements and their impact on pinout planning. For more information, see [Appendix A, Pinout Requirements](#).



Note 1: Pinout and clock region planning is required to accommodate regional clocking. See Appendix A, "Pinout Requirements," for details.

UG368\_c6\_11\_050809

Figure 6-11: 1 Gb/s RGMII Hewlett Packard v2.0 Clock Management

RGMII\_TXC is derived from the Ethernet MAC by routing GTX\_CLK to an IOB DDR output register, followed by an IODELAY element and an OBUF (which is then connected to the PHY). The IODELAY element is used to generate 2 ns of skew required between RGMII\_TXC and RGMII\_TXD at the FPGA device pads. This delay is specified in the *Hewlett Packard RGMII Specification, v2.0* to provide setup and hold times on the external interface.

RGMII\_RXC is generated by the PHY and connected to PHYEMACRXCLK via an IBUFG, BUFIO, and BUFR. Fixed-mode IODELAYs are instantiated on the RGMII clock and data lines. These are set to sample a 1 ns setup, 1 ns hold window at the device pads. RGMII\_RXC is buffered onto a regional clock net and drives all receive logic.

The CLIENTEMACDCMLOCKED port must be tied High.

## RGMII Standard Clock Management for Tri-Speed Operation

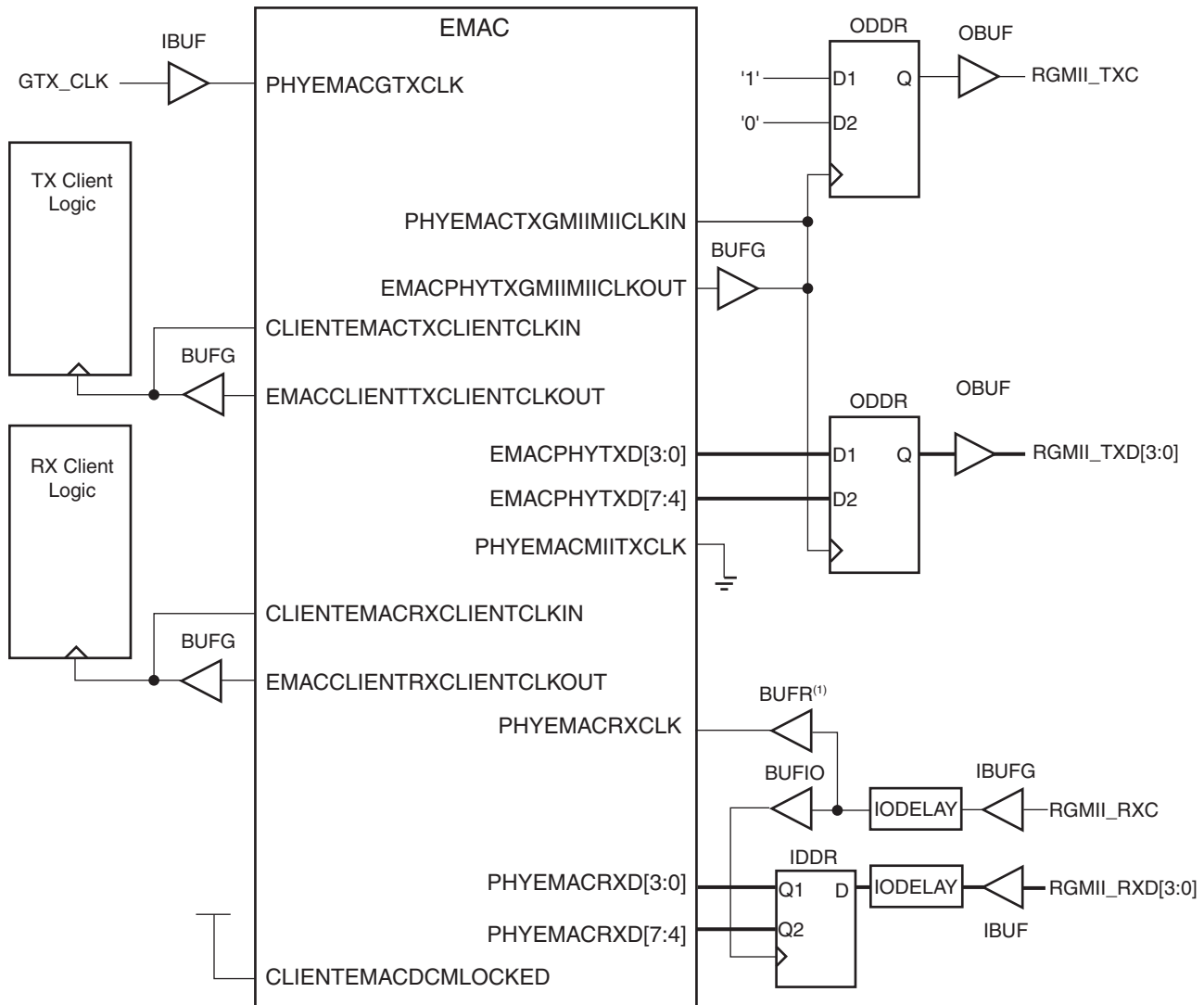
### RGMII Version 1.3

[Figure 6-12](#) shows the tri-speed clock management following the *Hewlett Packard RGMII Specification v1.3*. GTX\_CLK must be provided to the Ethernet MAC with a high-quality, 125 MHz clock that satisfies the IEEE Std 802.3-2005 requirements.

When using this clocking methodology, which routes the RGMII\_RXC net through a BUFIO and a BUFR, it is important to understand regional clocking requirements and their impact on pinout planning. For more information, see [Appendix A, Pinout Requirements](#).

The EMACPHYTXGMIIMICLKOUT port generates the appropriate frequency derived from GTX\_CLK and depending on the operating frequency of the link. It is routed through a BUFG and directly clocks the RGMII\_TXD ODDR registers.

A regional buffer (BUFR) can replace this BUFG. Refer to the Virtex-6 FPGA User Guide for BUFR usage guidelines.



Note 1: Pinout and clock region planning is required to accommodate regional clocking. See Appendix A, "Pinout Requirements," for details.

UG368\_c6\_12\_051409

Figure 6-12: Tri-Mode RGMII v1.3 Clock Management

The EMACCLIENTTXCLIENTCLKOUT output port connects to the CLIENTEMACTXCLIENTCLKIN input port and transmitter client logic in the FPGA logic through a BUFG. The receiver client clocking is similar.

RGMII\_TXC is derived from the Ethernet MAC by routing to an IOB DDR output register followed by an OBUF (which is then connected to the PHY). The use of the DDR register ensures that the forwarded clock is exactly in line with the RGMII transmitter data, as specified in the *Hewlett Packard RGMII Specification, v1.3*.

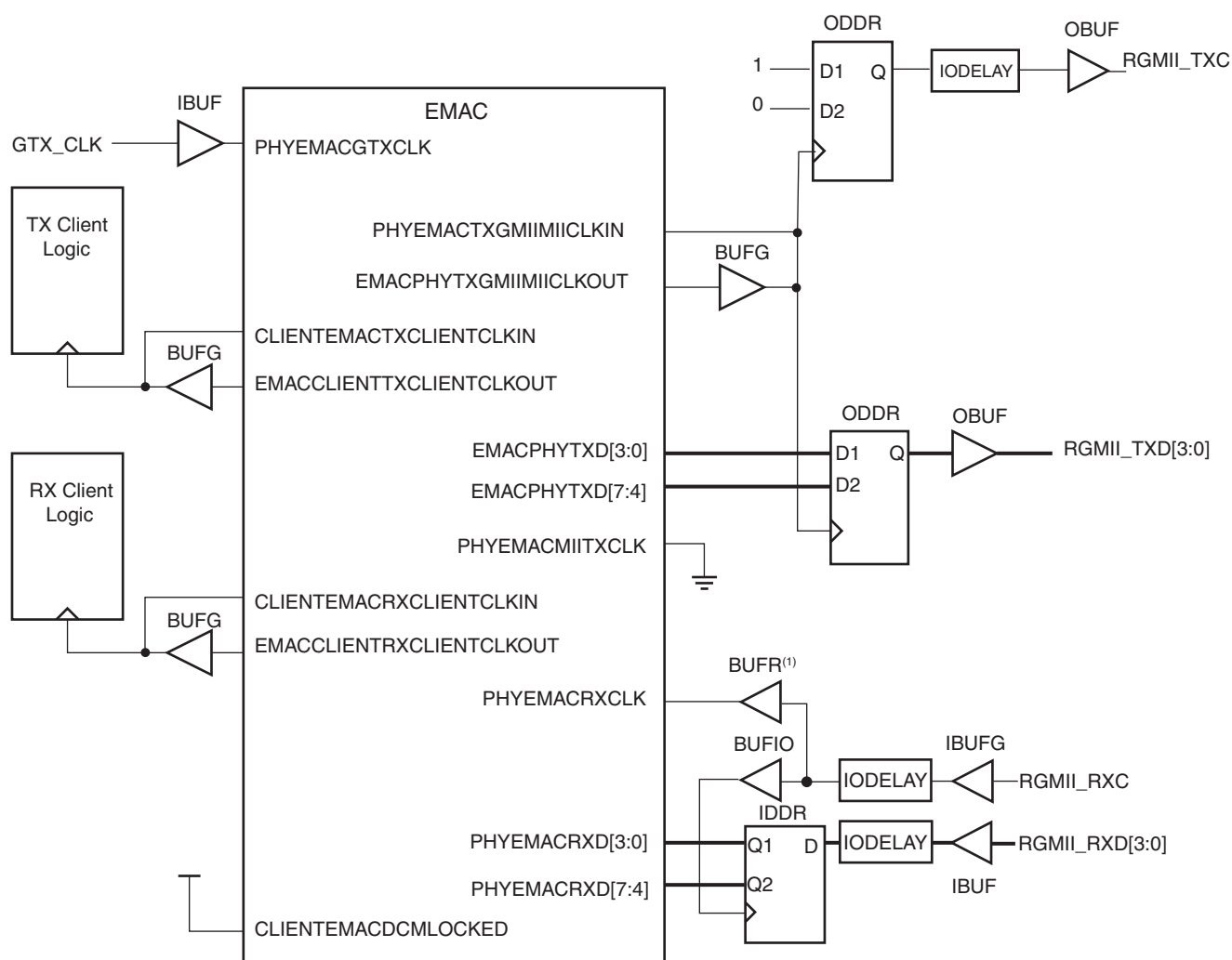
RGMII\_RXC is generated by the PHY and connected to PHYEMACRXCLK via an IBUFG and BUFR. A BUFIO is used to register the RGMII\_RXD inputs, and fixed-mode IODELAYS are instantiated on the RGMII clock and data lines. These are set to sample a

1 ns setup, 1 ns hold window at the device pads. The CLIENTEMACDCMLOCKED port must be tied High.

## RGMII Version 2.0

Figure 6-13 shows the tri-speed clock management following the *Hewlett Packard RGMII Specification v2.0*. GTX\_CLK must be provided to the Ethernet MAC with a high-quality, 125 MHz clock that satisfies the IEEE Std 802.3-2005 requirements. The EMACPHYTXGMIIMICLKOUT port generates the appropriate frequency deriving from GTX\_CLK and the operating frequency of the link. It is routed through a BUFG and directly clocks to the RGMII\_TXD ODDR registers. A regional buffer (BUFR) can replace this BUFG. Refer to the Virtex-6 FPGA User Guide for BUFR usage guidelines.

When using this clocking methodology, which routes the RGMII\_RXC net through a BUFIO and a BUFR, it is important to understand regional clocking requirements and their impact on pinout planning. For more information, see [Appendix A, Pinout Requirements](#).



Note 1: Pinout and clock region planning is required to accommodate regional clocking. See Appendix A, "Pinout Requirements," for details.

UG368\_c6\_13\_051409

Figure 6-13: Tri-Mode RGMII v2.0 Clock Management



RGMII\_TXC is derived from the Ethernet MAC by routing the EMACPHYTXGMIIIMCLKOUT port to an IOB DDR output register, followed by an IODELAY element and an OBUF (which is then connected to the PHY). The IODELAY element is used to generate 2 ns of skew required between RGMII\_TXC and RGMII\_TXD at the FPGA device pads. This delay is specified in the *Hewlett Packard RGMII Specification, v2.0* to provide setup and hold times on the external interface.

The EMACCLIENTTXCLIENTCLKOUT output port connects to the CLIENTEMACTXCLIENTCLKIN input port and transmitter client logic in the FPGA logic through a BUFG. The receiver client clocking is similar.

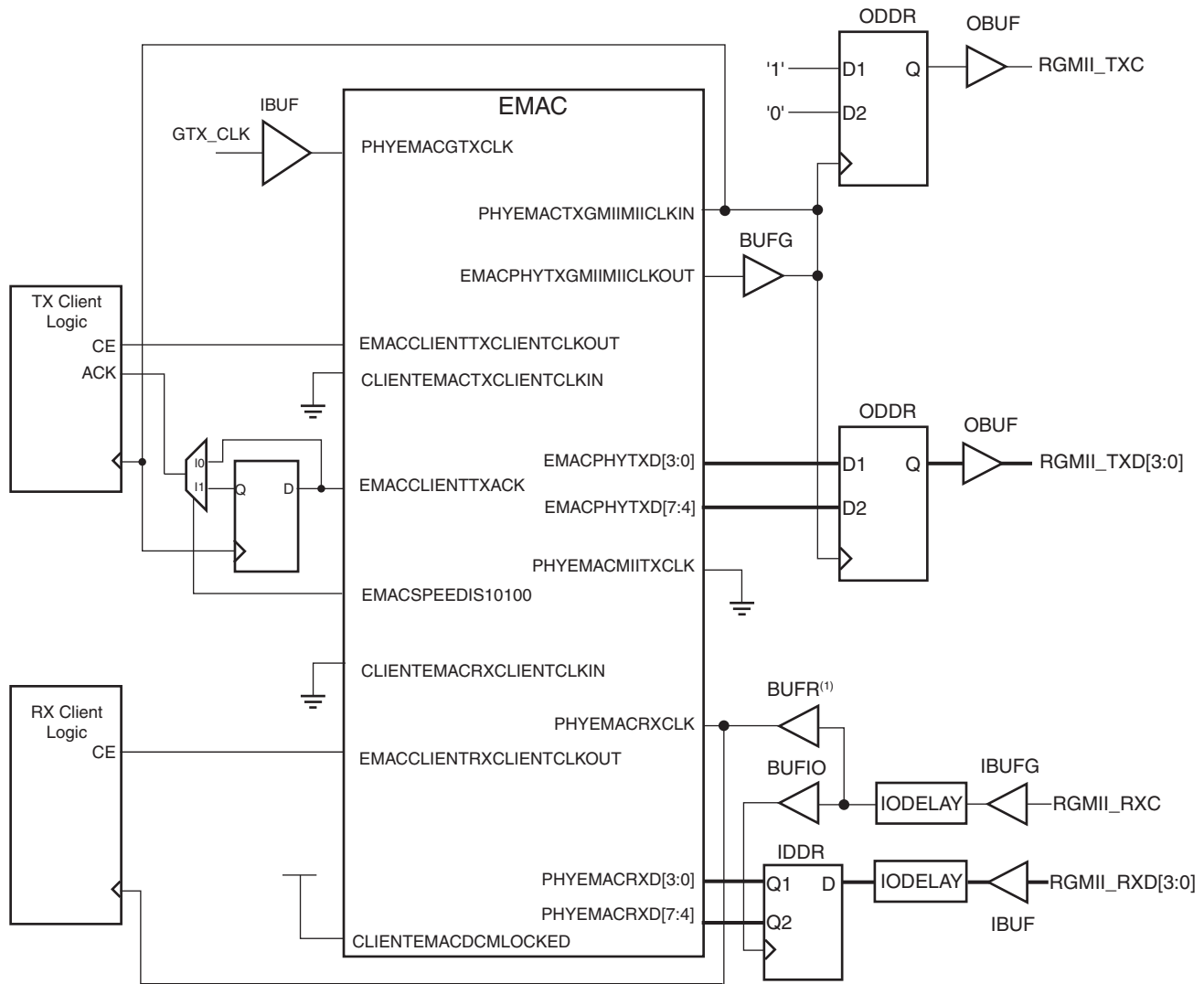
RGMII\_RXC is generated by the PHY and connected to PHYEMACRXCLK via an IBUFG and BUFR. A BUFIO is used to register the RGMII\_RXD inputs, and fixed-mode IODELAYs are instantiated on the RGMII clock and data lines. These are set to sample a 1 ns setup, 1 ns hold window at the device pads. The CLIENTEMACDCMLOCKED port must be tied High.

## RGMII Clock Management for Tri-Speed Operation Using Clock Enables

### RGMII Version 1.3

[Figure 6-14](#) shows the clock management scheme for tri-speed RGMII v1.3 operation when the EMAC\_USECLKEN attribute is set.

When using this clocking methodology, which routes the RGMII\_RXC net through a BUFIO and a BUFR, it is important to understand regional clocking requirements and their impact on pinout planning. For more information, see [Appendix A, Pinout Requirements](#).



Note 1: Pinout and clock region planning is required to accommodate regional clocking. See Appendix A, "Pinout Requirements," for details.

UG368\_c6\_14\_051109

**Figure 6-14: Tri-Mode RGMII with Clock Enables**

In this mode of operation, the transmitter signals are synchronous to PHYEMACTXGMIIICLKIN, and the receiver signals are synchronous to PHYEMACRXCLK.

At 1 Gb/s, all external logic is clocked at 125 MHz. At 100 Mb/s and 10 Mb/s, the logic is clocked at 25 MHz and 2.5 MHz, respectively. To maintain the correct data rate at the client interface, clock enable signals are output on EMACCLIENTTXCLIENTCLKOUT for the transmitter logic, and EMACCLIENTRXCLIENTCLKOUT for the receiver logic. These signals are High during 1 Gb/s operation and toggle on each clock edge at slower speeds. These are used to enable the client interface logic.

Due to the timing relationship between the RGMII transmit clock and the internal client clock signals, the TX\_ACK signal (EMACCLIENTTXACK) is registered at the output of the Ethernet MAC when operating at speeds below 1 Gb/s. At 1 Gb/s, the register is bypassed.

RGMII\_TXC is derived from the Ethernet MAC by routing to an IOB DDR output register followed by an OBUF (which is then connected to the PHY). The use of the DDR register ensures that the forwarded clock is exactly in line with the RGMII transmitter data, as specified in the *Hewlett Packard RGMII Specification, v1.3*.

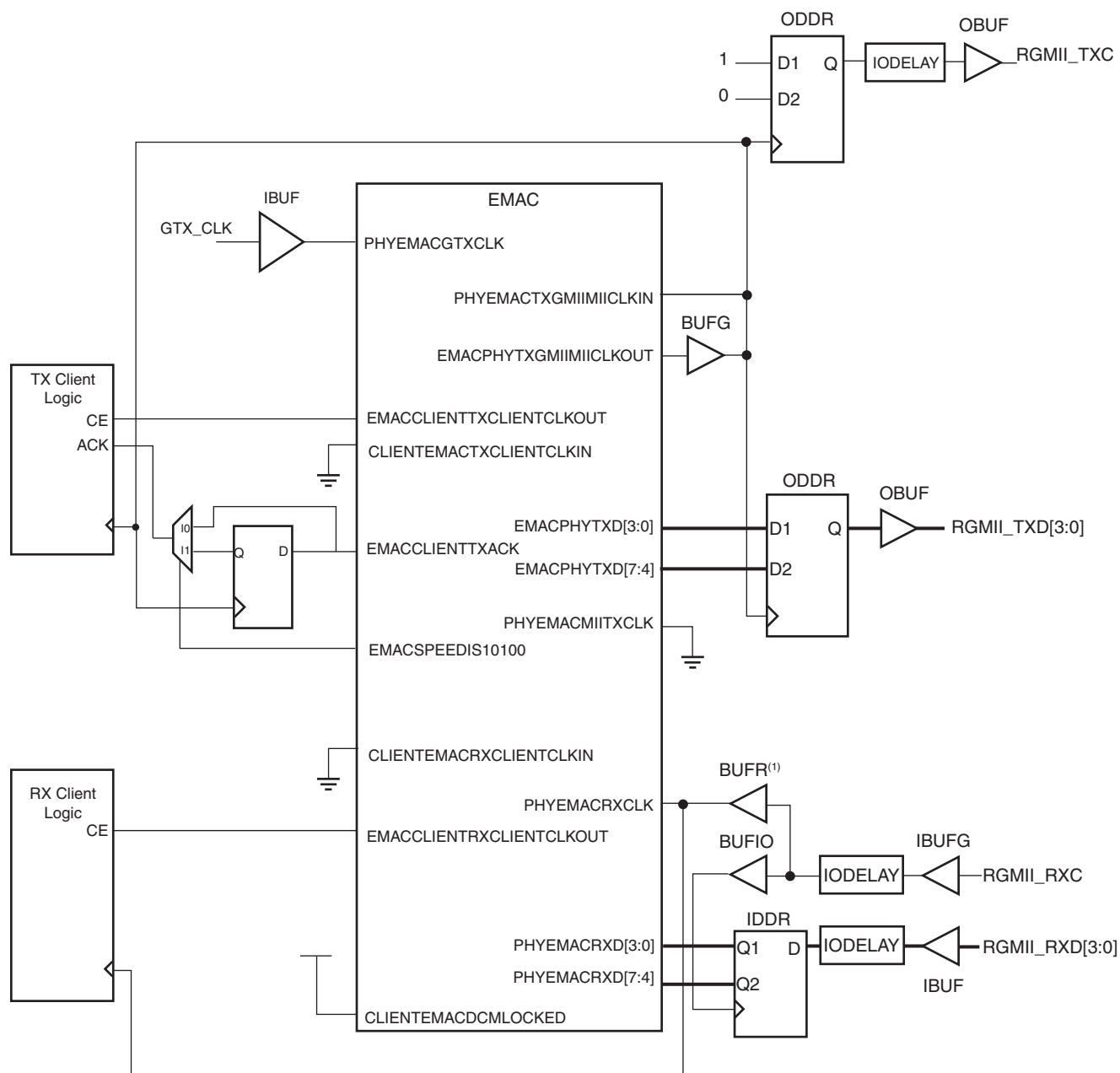
RGMII\_RXC is generated by the PHY and connected to PHYEMACRXCLK via an IBUFG and BUFR. This regional clock net is also used to clock FPGA receiver logic, including the receiver client. A BUFIO is used to register the RGMII\_RXD inputs, and fixed-mode IODELAYs are instantiated on the RGMII clock and data lines. These are set to sample a 1 ns setup, 1 ns hold window at the device pads. The CLIENTEMACDCMLOCKED port must be tied High.

This clocking scheme requires two fewer BUFs than the default scheme detailed in [Figure 6-12](#).

## RGMII Version 2.0

Figure 6-15 shows the clock management scheme for tri-speed RGMII v2.0 operation when the EMAC\_USECLKEN attribute is set.

When using this clocking methodology, which routes the RGMII\_RXC net through a BUFIO and a BUFR, it is important to understand regional clocking requirements and their impact on pinout planning. For more information, see [Appendix A, Pinout Requirements](#).



Note 1: Pinout and clock region planning is required to accommodate regional clocking. See Appendix A, "Pinout Requirements," for details.

UG368\_c6\_15\_050809

Figure 6-15: Tri-Mode RGMII with Clock Enables

In this mode of operation, the transmitter signals are synchronous to PHYEMACTXGMIIIMCLKIN, and the receiver signals are synchronous to PHYEMACRXCLK.

At 1 Gb/s, all external logic is clocked at 125 MHz. At 100 Mb/s and 10 Mb/s, the logic is clocked at 25 MHz and 2.5 MHz, respectively. To maintain the correct data rate at the client interface, clock enable signals are output on EMACCLIENTTXCLIENTCLKOUT for the transmitter logic, and EMACCLIENTRXCLIENTCLKOUT for the receiver logic. These signals are High during 1 Gb/s operation and toggle on each clock edge at slower speeds. These are used to enable the client interface logic.

Due to the timing relationship between the RGMII transmit clock and the internal client clock signals, the TX\_ACK signal (EMACCLIENTTXACK) is registered at the output of the Ethernet MAC when operating at speeds below 1 Gb/s. At 1 Gb/s, the register is bypassed.

RGMII\_TXC is derived from the Ethernet MAC by routing the EMACPHYTXGMIIIMCLKOUT port to an IOB DDR output register, followed by an IODELAY element and an OBUF (which is then connected to the PHY). The IODELAY element is used to generate 2 ns of skew required between RGMII\_TXC and RGMII\_TXD at the FPGA device pads. This delay is specified in the *Hewlett Packard RGMII Specification, v2.0* to provide setup and hold times on the external interface.

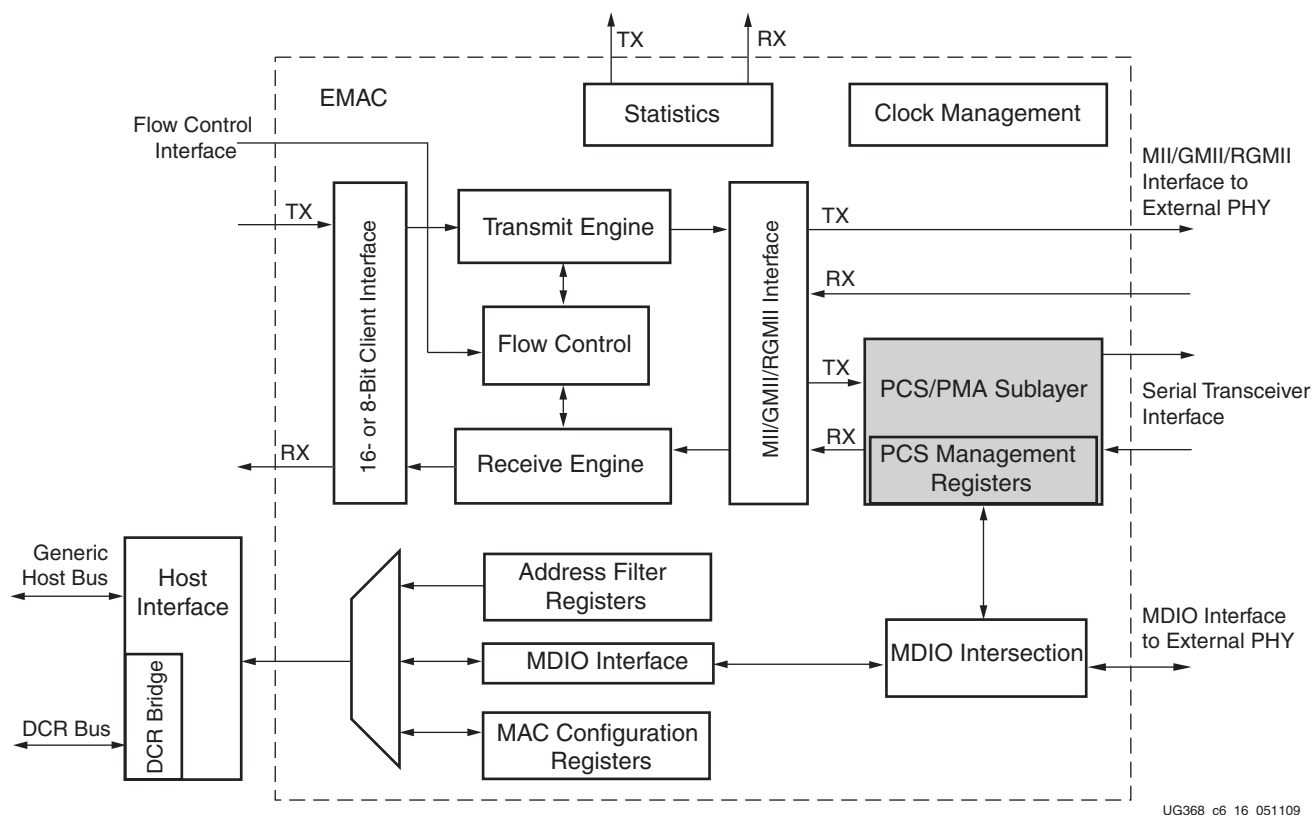
RGMII\_RXC is generated by the PHY and connected to PHYEMACRXCLK via an IBUFG and BUFR. This regional clock net is also used to clock FPGA receiver logic, including the receiver client. A BUFIO is used to register the RGMII\_RXD inputs, and fixed-mode IODELAYs are instantiated on the RGMII clock and data lines. These are set to sample a 1 ns setup, 1 ns hold window at the device pads. The CLIENTEMACDCMLOCKED port must be tied High.

This clocking scheme requires two fewer BUFs than the default scheme detailed in [Figure 6-13](#).

## 1000BASE-X PCS/PMA

## Ethernet MAC PCS/PMA Sublayer

Figure 6-16 shows the functional block diagram of the Ethernet MAC with the PCS/PMA sublayer block highlighted.



**Figure 6-16: Functional Block Diagram of the Ethernet MAC**

The PCS/PMA sublayer extends the functionality of the Ethernet MAC, replacing the GMII/MII or RGMII parallel interfaces with an interface to a serial transceiver. The connected serial transceiver then provides the remaining functionality to accommodate the following two standards, both of which require the high-speed serial interface provided by the serial transceiver:

- Serial Gigabit Media Independent Interface (SGMII)
- 1000BASE-X PCS/PMA

# Introduction to the 1000BASE-X PCS/PMA Implementation

The 1000BASE-X physical standard, described in IEEE Std 802.3-2005, clauses 36 and 37, defines a physical sublayer that is usually connected to an optical fiber medium. Two common implementations currently exist: 1000BASE-LX and 1000BASE-SX (long and short wavelength laser), which can both be obtained by connecting the serial transceiver to a suitable GBIC or SFP optical transceiver.

Figure 6-17 shows an expanded diagram of the PCS/PMA sublayer block. The PCS/PMA sublayer contains its own functional blocks, which are illustrated and summarized in the following subsections. The serial transceiver is shown connected to an external optical transceiver to complete the 1000BASE-X optical link.

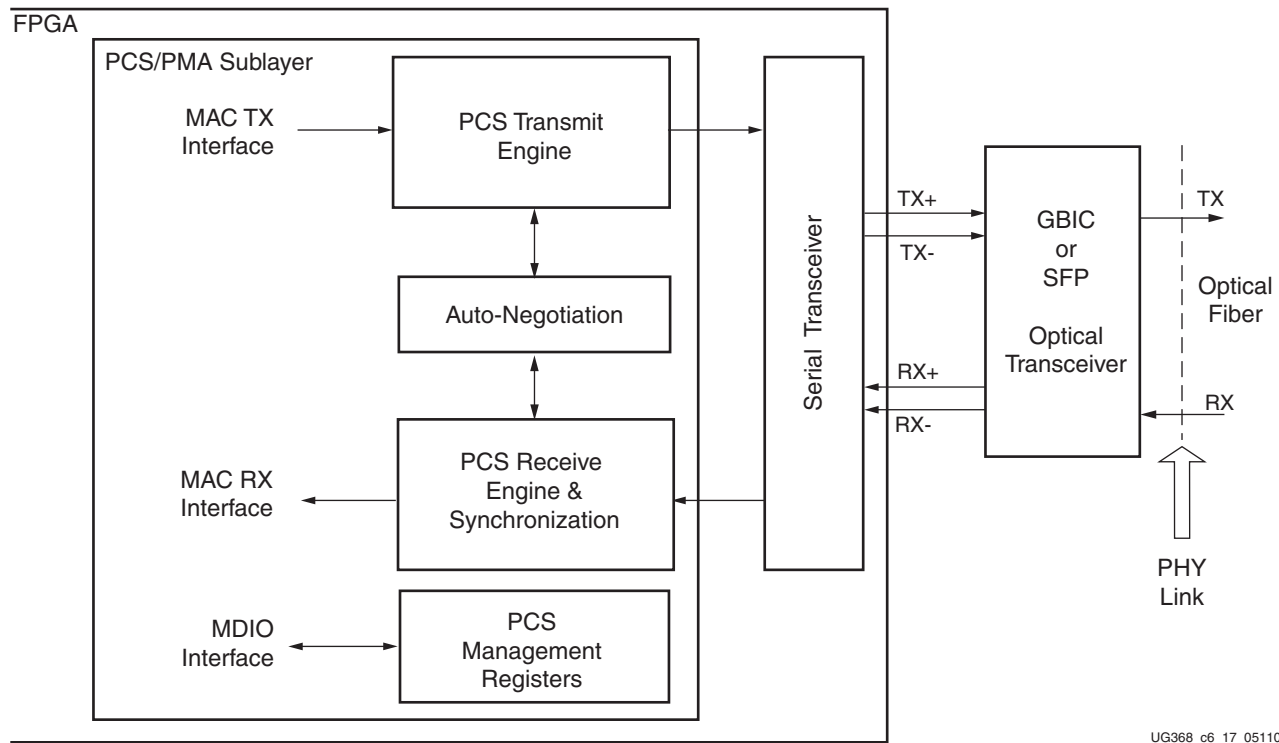


Figure 6-17: Ethernet PCS/PMA Sublayer Extension

The PCS Transmit Engine encodes the datastream received from the MAC into a sequence of ordered sets as defined by the standard. The data is transmitted to the serial transceiver using an 8-bit datapath that then provides 8B/10B encoding of this data and parallel to serial conversion. The output from the serial transceiver is a differential pair operating at a rate of 1.25 Gb/s.

For more information on the PCS Transmit Engine, refer to the state diagrams of IEEE Std 802.3-2005 (Figures 36-5 and 36-6).

## PCS Receive Engine and Synchronization

The serial transceiver receives a serial differential pair operating at a rate of 1.25 Gb/s from the optical transceiver. The serial transceiver, using CDR, performs word alignment on this serial datastream and then converts this to a parallel interface. This data is then 8B/10B decoded by the serial transceiver before presenting this to the PCS/PMA sublayer using an 8-bit datapath.

Within the PCS/PMA sublayer, the synchronization process performs analysis of valid/invalid received sequence ordered sets to determine if the link is in good order. The PCS receive engine then decodes these sequence ordered sets into a format that can then be presented to the standard receiver datapath within the Ethernet MAC.

For more information on the synchronization process, refer to IEEE Std 802.3-2005 (Figure 36-9). For the PCS Receive Engine, refer to IEEE Std 802.3-2005 (Figures 36-7a and 36-7b).

## Auto-Negotiation

The 1000BASE-X auto-negotiation function allows a device to advertise the supported modes of operation to a device at the remote end of the optical fiber (the link partner) and to detect corresponding operational modes that the link partner advertises.

Auto-negotiation is controlled and monitored using a software function through the PCS Management Registers. See [1000BASE-X Auto-Negotiation](#), page 164.

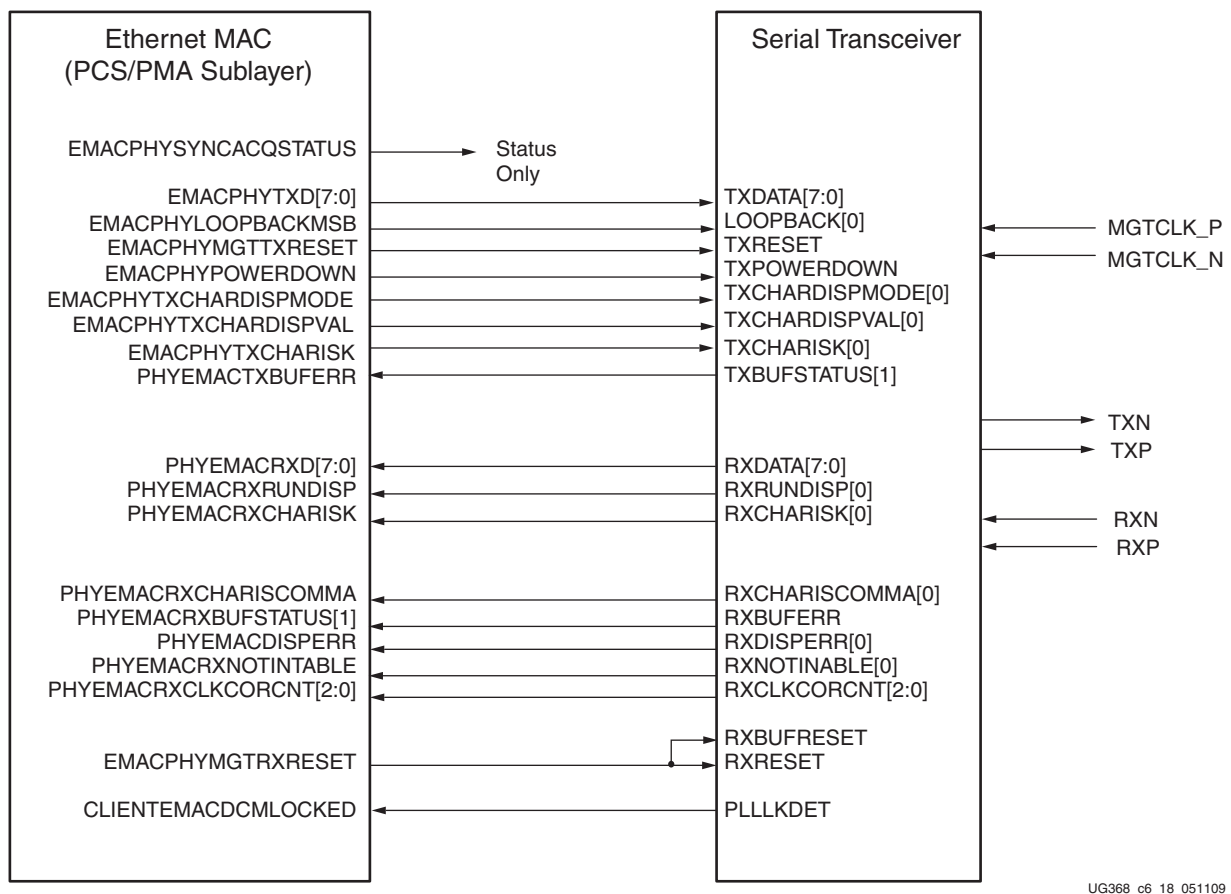
## PCS Management Registers

Configuration and status of the PCS/PMA sublayer, including access to and from the auto-negotiation function, is via the PCS Management Registers. These registers are accessed through the serial Management Data Input/Output Interface (MDIO), as shown in [Chapter 5, MDIO Interface](#).

For the 1000BASE-X standard, the configuration and status registers available are listed in [1000BASE-X PCS/PMA Management Registers](#), page 118.

## Ethernet MAC to Serial Transceiver Connections

[Figure 6-18](#) shows the Ethernet MAC configured with 1000BASE-X PCS/PMA as the physical interface. Connections to the Virtex-6 FPGA serial transceiver are illustrated.



UG368\_c6\_18\_051109

Figure 6-18: Ethernet MAC Configured in 1000BASE-X PCS/PMA Mode



These connections are created by the CORE Generator tool when the physical interface is selected to be either SGMII or 1000BASE-X PCS/PMA. By using the CORE Generator tool, the time required to instantiate the Ethernet MAC into a usable design is greatly reduced. See [Accessing the Ethernet MAC from the CORE Generator Tool](#), page 21.

## 1000BASE-X PCS/PMA Clock Management

### 8-Bit Data Client

Figure 6-19 shows the clock management used with the 1000BASE-X PCS/PMA interface when the Ethernet MAC client is used with an 8-bit client data width.

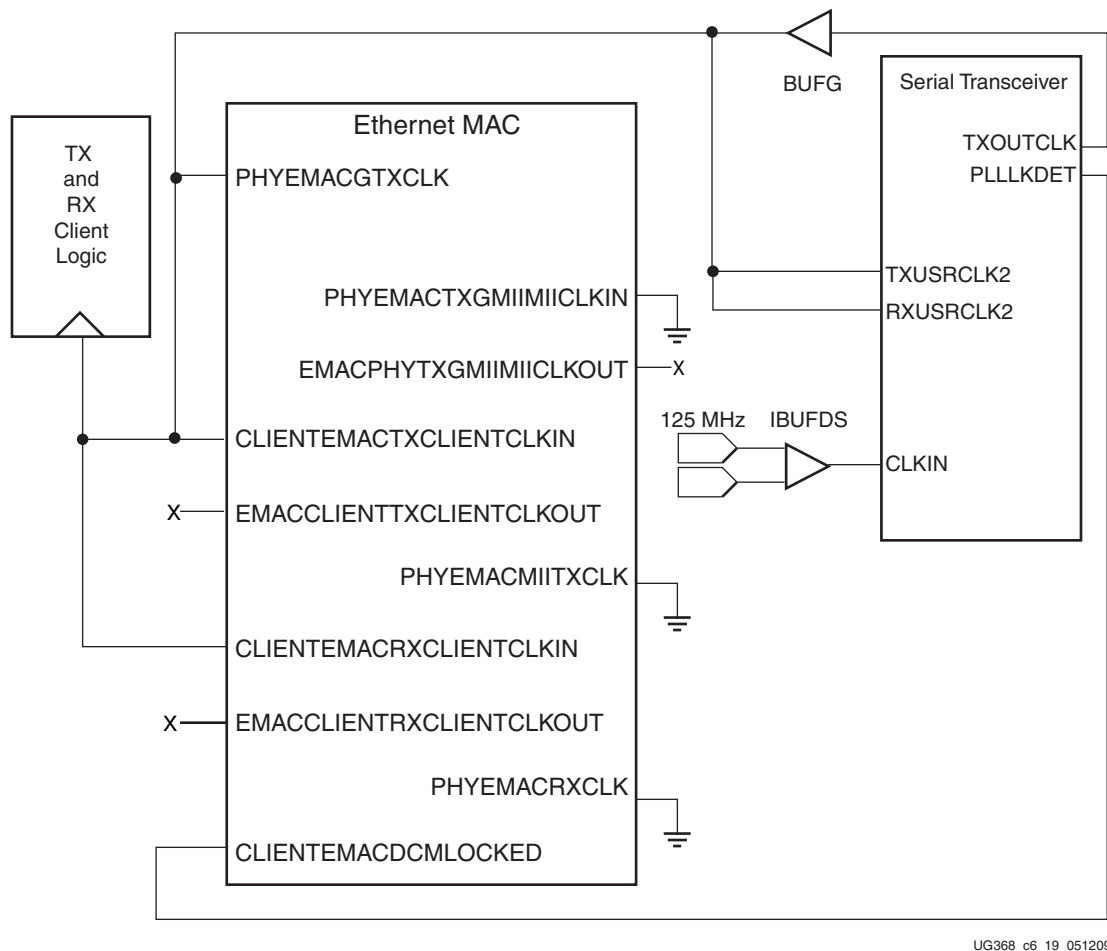


Figure 6-19: 1000BASE-X PCS/PMA (8-Bit Data Client) Clock Management

The CLKIN inputs to the serial transceiver must be connected to an external, high-quality differential reference clock of frequency of 125 MHz. A 125 MHz clock source is then provided to the FPGA logic from the TXOUTCLK output port of the serial transceiver. This is connected to global clock routing using a BUFG as illustrated in Figure 6-19. This clock should then be routed to the PHYEMACGTCLK port of the Ethernet MAC.

Additionally, this global clock can be used for both receiver and transmitter client logic, as illustrated in Figure 6-19, and it therefore must be routed back to the Ethernet MAC

through the CLIENTEMACRXCLIENTCLKIN and CLIENTEMACTXCLIENTCLKIN input ports.

The PLLKDET signal from the serial transceiver (indicating that its internal PLLs have locked) is routed to the CLIENTEMACDCMLOCKED input port of the Ethernet MAC. This ensures that the state machines of the Ethernet MAC are held in reset until the serial transceiver has locked, and its clocks are running cleanly.

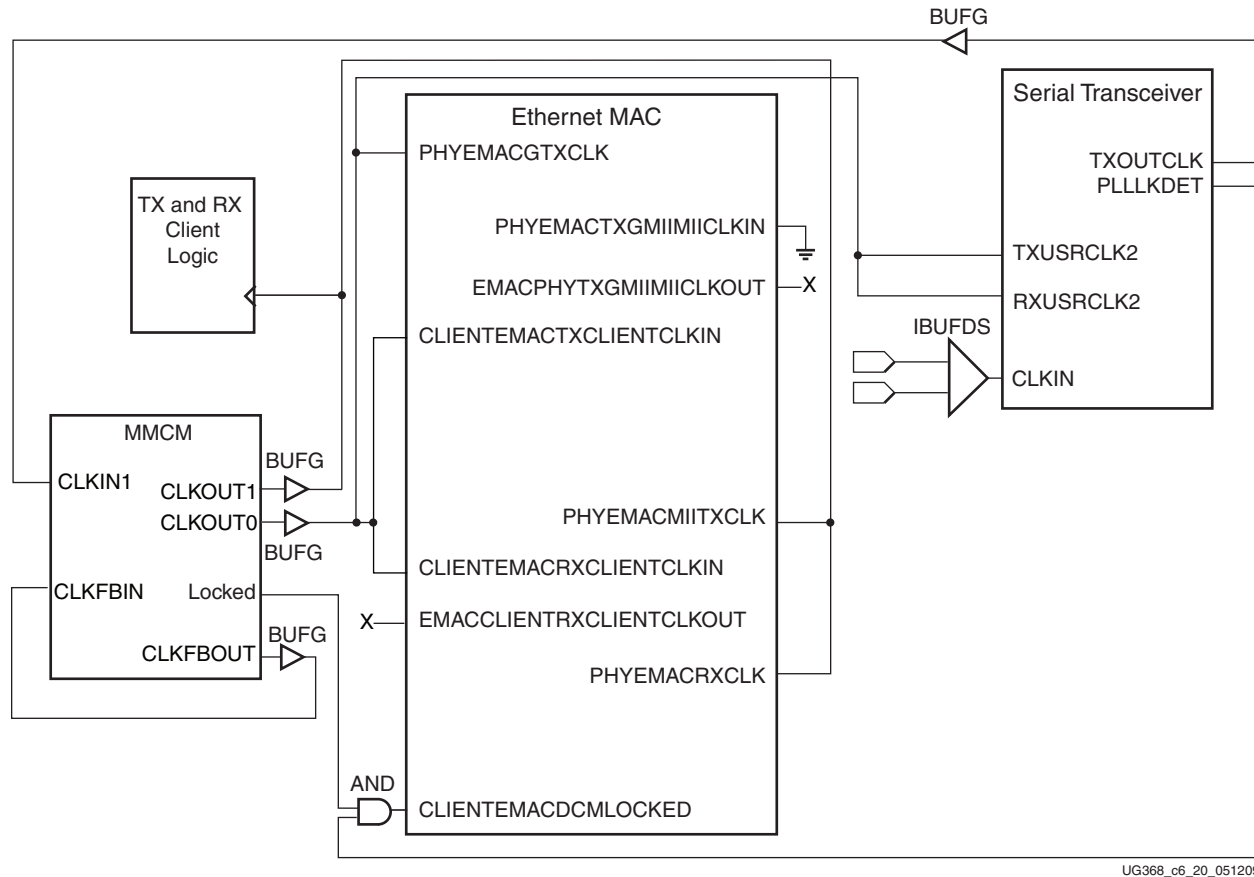
As described in [Appendix B, Ethernet MAC Clocks](#), the following clock signals are unused:

- EMACCLIENTRXCLIENTCLKOUT
- EMACCLIENTTXCLIENTCLKOUT
- EMACPHYTXGMIIIMICLKOUT
- PHYEMACTXGMIIIMICLKIN
- PHYEMACMIITXCLK
- PHYEMACRXCLK

The outputs can be left unconnected, and inputs should be tied to Low.

## 16-Bit Data Client

[Figure 6-20](#) shows the clock management used with the 1000BASE-X PCS/PMA interface and a 16-bit client data width. This mode supports overclocking up to a data rate of 2.5 Gb/s based upon the combination of the serial transceiver reference clock frequency and MMCM attributes. Because IEEE Std 802.3-2005 specifies a 1000BASE-X rate of 1 Gb/s, overclocking should not be used for Ethernet-compliant designs; however, it can be useful for some backplane implementations.



**Figure 6-20: 1000BASE-X PCS/PMA (16-Bit Data Client) Clock Management**

The CLKIN inputs to the serial transceiver must be connected to an external, high-quality, differential reference clock with a frequency as listed in [Table 6-1](#).

A clock source matching the reference clock frequency is then provided to the FPGA logic from the TXOUTCLK output port of the serial transceiver. This clock is then routed to the CLKIN1 input of an MMCM. CLKOUT1 of the MMCM is connected to global clock routing using a BUFG, as illustrated in [Figure 6-20](#). The output of the BUFG is connected to the PHYEMACMIITXCLK and PHYEMACRXCLK ports of the Ethernet MAC. It is also used to clock both receiver and transmitter logic for the Ethernet MAC 16-bit client interface.

The global clock sourced from the MMCM's CLKOUT0 port, running at twice the frequency of CLKOUT1, must be routed back to the Ethernet MAC through the input ports CLIENTEMACRXCLIENTCLKIN and CLIENTEMACTXCLIENTCLKIN. This clock is also connected to the PHYEMACGTXCLK input of the Ethernet MAC and the RXUSRCLK2 and TXUSRCLK2 inputs of the transceiver.

The PLLLKDET signal from the serial transceiver (indicating that its internal PLLs have locked) is ANDed with the locked signal from the MMCM and routed to the CLIENTEMACDCMLOCKED input port of the Ethernet MAC. This ensures that the state machines of the Ethernet MAC are held in reset until the serial transceiver has locked and all clocks are running cleanly.

**Table 6-1: Serial Transceiver and Example MMCM Configurations to Support 16-bit Client Interface**

	1.0 Gb/s	2.0 Gb/s	2.5 Gb/s
Serial Transceiver Settings			
Reference clock frequency (CLKIN port)	125 MHz	125 MHz	156.25 MHz
Example MMCM Attributes			
CLKIN1_PERIOD	8.0	8.0	6.4
CLKFBOUT_MULT_F	5.0	5.0	5.0
CLKOUT0_DIVIDE_F	5.0	2.5	2.5
CLKOUT1_DIVIDE	10	5	5

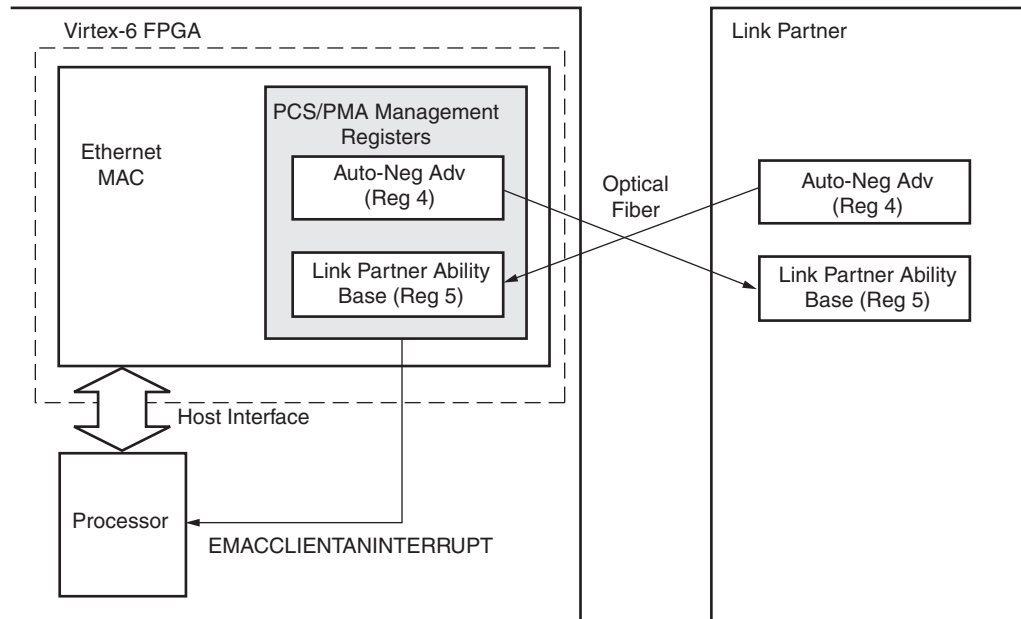
As described in [Appendix B, Ethernet MAC Clocks](#), the following clock signals are unused and can be left unconnected:

- EMACCLIENTRXCLIENTCLKOUT
- EMACPHYTXGMIIMICLKOUT
- EMACCLIENTTXCLIENTCLKOUT

## 1000BASE-X Auto-Negotiation

### Overview of Operation

[Figure 6-21](#) illustrates a simplified diagram of the Ethernet MAC instantiated within a Virtex-6 device. The only components shown are two of the PCS Management Registers that are directly involved in the auto-negotiation process (see [1000BASE-X PCS/PMA Management Registers, page 118](#)). The corresponding registers of the connected device is also shown.



UG368\_c6\_21\_051109

Figure 6-21: 1000BASE-X Auto-Negotiation Overview

IEEE Std 802.3-2005, clause 37 describes the 1000BASE-X auto-negotiation function. This function allows a device to advertise its supported modes of operation to a device at the remote end of a link segment (the link partner) and to detect corresponding operational modes advertised by the link partner. The operation of the 1000BASE-X auto-negotiation is summarized below:

- To enable auto-negotiation, both auto-negotiation (see [Table 5-3, page 119](#)) and MDIO (see [Table 4-8, page 88](#)) must be enabled. If enabled, auto-negotiation starts automatically:
  - After power-up/reset
  - Upon loss of synchronization
  - Whenever the link partner initiates auto-negotiation
  - Whenever an auto-negotiation restart is requested (see [Table 5-2, page 119](#))
  - When the PHY Reset bit is set in the PCS control register (see [Table 5-2, page 119](#))
- During auto-negotiation, the contents of the [Table 5-6, page 121](#) are transferred to the link partner. This register can be written to through the management interface, and enables software control of the systems advertised abilities. Information provided in this register includes:
  - Fault condition signalling
  - Duplex mode
  - Flow control capabilities for the Ethernet MAC
- At the same time, the advertised abilities of the link partner are transferred into the [Table 5-7, page 122](#). This includes the same information fields as in the [Table 5-6, page 121](#).

- Under normal conditions, this completes the auto-negotiation information exchange. The results can be read from the [Table 5-7, page 122](#). The mode of the Ethernet MAC should then be configured by a software routine to match; this does not happen automatically within the Ethernet MAC. The two methods by which a host processor can learn of the completion of an auto-negotiation cycle are:
  - By polling the auto-negotiation completion bit 1.5 in [Table 5-4, page 120](#)
  - By using the auto-negotiation interrupt port (see [Auto-Negotiation Interrupt](#)).

## 1000BASE-X Auto-Negotiation Link Timer

The auto-negotiation link timer is used to time three phases of the auto-negotiation procedure. For the 1000BASE-X standard, this link timer is defined as having a duration of between 10 and 20 ms. Both link partners wait until their own link timer and the partner's link timer expire before moving on to the next phase of the auto negotiation process. The complete process takes a minimum of three times the values of the biggest link timer value.

The duration of the link timer used by the Ethernet MAC can be configured with the `EMAC_LINKTIMERVAL[8:0]` attribute (see [Table 2-18, page 41](#)) or by writing to the appropriate host register (see [Table 4-6, page 86](#)). The duration of the timer is approximately equal to the binary value placed onto this attribute multiplied by  $32.768\ \mu\text{s}$  (4,096 clock periods of the 125 MHz clock provided to the Ethernet MAC on `PHYEMACGTCLK`). The accuracy of this link timer is within the range:

+0 to  $-32.768\ \mu\text{s}$

Therefore, for the 1000BASE-X standard, the attribute `EMAC_LINKTIMERVAL[8:0]` is set to:

`100111101` = 317 decimal

This setting corresponds to a timer duration of between 10.354 and 10.387 ms. This value can be reduced for simulation. The link timer value must be adjusted to accommodate for the change in reference clock frequency if operating the Ethernet MAC at an overclocked rate of 2500 Mb/s in 16-bit client interface mode.

## Auto-Negotiation Interrupt

The auto-negotiation function has an `EMACCLIENTANINTERRUPT` port. This port is designed to be used with common microprocessor bus architectures.

The operation of this port is enabled or disabled and cleared via the [Table 5-13, page 124](#)

- When disabled, this port is permanently driven Low.
- When enabled, this port is set to logic 1 following the completion of an auto-negotiation cycle. It remains High until cleared after a zero is written to bit 16.1 (Interrupt Status bit) of the [Table 5-12, page 124](#).

## Use of Clock Correction Sequences

The serial transceiver is configured by the Virtex-6 FPGA GTX Transceiver Wizard to perform clock correction. The output of the Transceiver Wizard is provided as part of the Ethernet MAC wrapper generated using the CORE Generator tool. Two different clock correction sequences can be employed:

- The mandatory clock correction sequence is the `/I2/` ordered set; this is a two-byte code-group sequence formed from `/K28.5/` and `/D16.2/` characters. The `/I2/` ordered set is present in the interframe gap. These sequences can therefore be

removed or inserted by the transceiver's receiver elastic buffer without causing frame corruption.

- The default Transceiver Wizard configuration enables the `CLK_COR_SEQ_2_USE` attribute. In this case, the transceiver is also configured to perform clock correction on the `/K28.5/D21.5/` sequence; these are the first two code-groups from the `/C1/` ordered set (the `/C1/` ordered set is four code-groups in length). Because there are no `/I2/` ordered sets present during much of the auto-negotiation cycle, this provides a method of allowing clock correction to be performed during auto-negotiation. Because this form of clock correction inserts or removes two code groups into or from a four code-group sequence, this causes ordered-set fragments to be seen by the Ethernet MAC's auto-negotiation state machine. It is therefore important that the transceiver's `RXCLKCORCNT[2:0]` port be correctly connected to the Ethernet MAC; this indicates a clock correction event (and type) to the Ethernet MAC. Using this signal, the Ethernet MAC's state machine can interpret the clock-correction fragments, and the auto-negotiation function can complete cleanly.

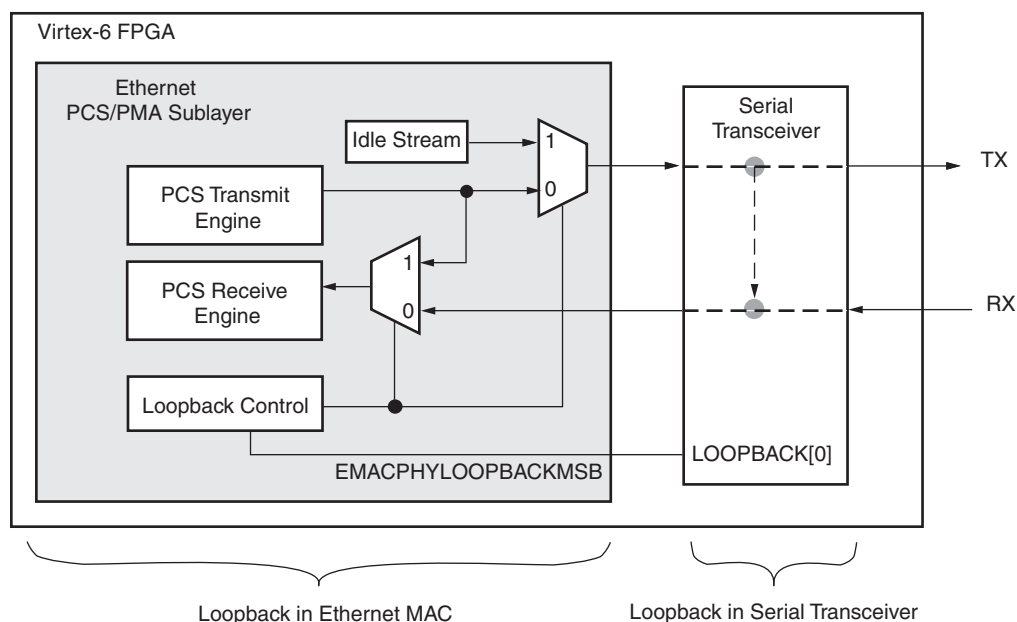
When the transceiver's `CLK_COR_SEQ_2_USE` attribute is not enabled, no clock correction can be performed during much of the auto-negotiation cycle. When this is the case, it is possible that the transceiver's receiver elastic buffer could underflow or overflow as asynchronous clock tolerances accumulate. This results in an elastic buffer error. It is therefore important that the transceiver's `RXBUFSTATUS[2:0]` port is correctly connected to the Ethernet MAC; this indicates a buffer error event to the Ethernet MAC. Using this signal, the Ethernet MAC's state machine can interpret the buffer error and the auto-negotiation function can complete cleanly.

The serial transceiver is by default configured to perform clock correction during the auto-negotiation cycle. If correctly connected as per the example design, the Ethernet MAC's state machine can correctly determine the transceiver's elastic buffer behavior, and auto-negotiation can complete cleanly.

## Loopback When Using the PCS/PMA Sublayer for 1000BASE-X

Figure 6-22 illustrates the loopback options when using the PCS/PMA sublayer. Two possible loopback positions are shown:

- Loopback in the Ethernet MAC.** When placed into loopback, data is routed from the transmitter to the receiver path at the last possible point in the PCS/PMA sublayer. This is immediately before the serial transceiver interface. When placed into loopback, a constant stream of Idle code groups are transmitted through the serial transceiver. Loopback in this position allows test frames to be looped back within the Ethernet MAC without allowing them to be received by the link partner (the device connected at the other end of the optical link). The transmitting of idles allows the link partner to remain in synchronization so that no fault is reported.
- Loopback in the serial transceiver.** The serial transceiver can alternatively be switched into loopback by connecting the EMACPHYLOOPBACKMSB port of the Ethernet MAC to the loopback port of the serial transceiver as illustrated. The serial transceiver loopback routes data from the transmitter path to the receiver path, within the serial transceiver. However, this data is also transmitted out of the serial transceiver and so any test frames used for a loopback test are received by the link partner.



UG368\_c6\_22\_051109

Figure 6-22: Loopback When Using the PCS/PMA Sublayer

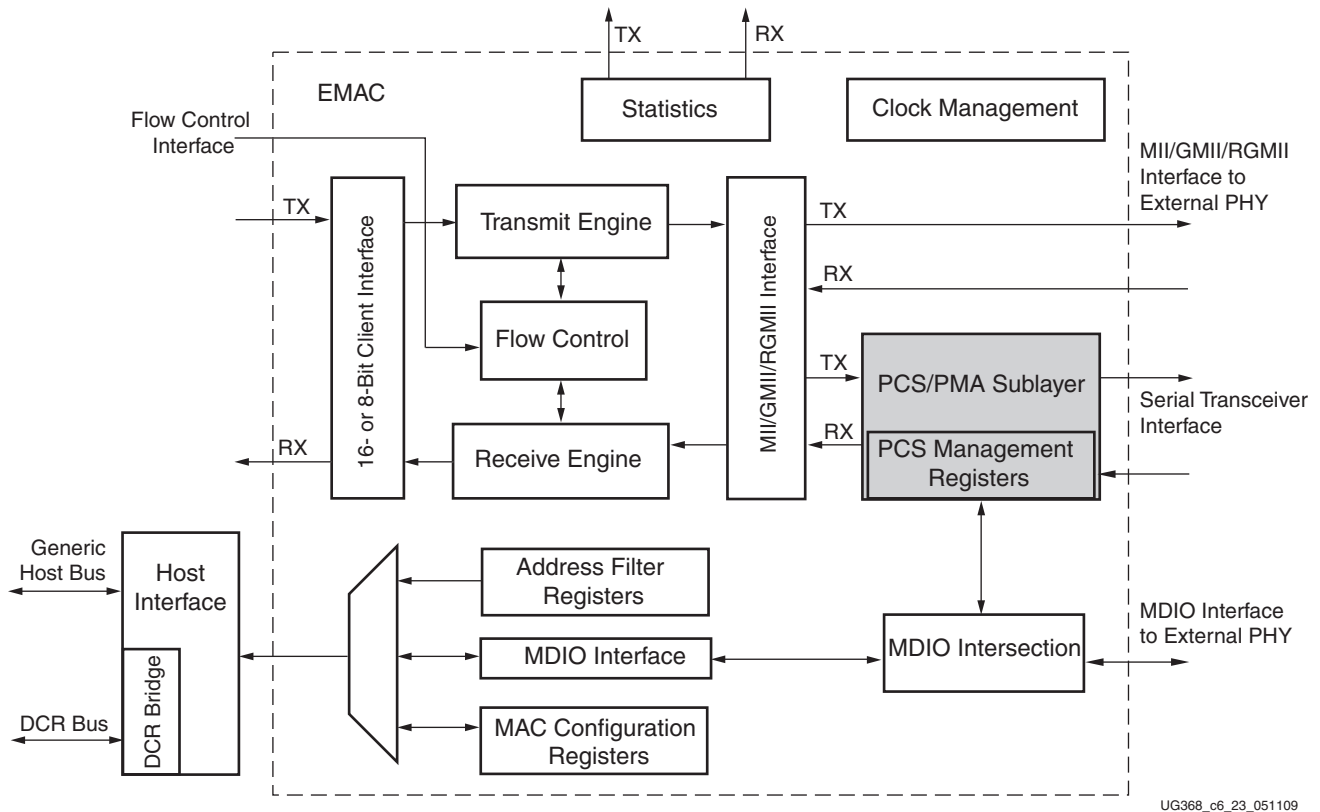
Loopback itself can be enabled or disabled by writing to the PCS/PMA Sublayer Control Register (Register 0). The loopback position can be controlled through the Loopback Control Register (Register 17). See [1000BASE-X PCS/PMA Management Registers, page 118](#) for details. Alternatively, loopback can be controlled by setting the EMAC\_PHYLOOPBACKMSB and EMAC\_GTLOOPBACK attributes. See [Chapter 2](#) for details.



# Serial Gigabit Media Independent Interface (SGMII)

## Ethernet MAC PCS/PMA Sublayer

Figure 6-23 shows the functional block diagram of the Ethernet MAC with the PCS/PMA sublayer block highlighted.



**Figure 6-23: PCS/PMA Sublayer Block Diagram in the Ethernet MAC**

The PCS/PMA sublayer extends the functionality of the Ethernet MAC, replacing the GMII/MII or RGMII parallel interfaces with an interface to a serial transceiver. The connected serial transceiver then provides the remaining functionality to accommodate the following two standards, both of which require the high-speed serial interface provided by the serial transceiver:

- Serial Gigabit Media Independent Interface (SGMII)
- 1000BASE-X PCS/PMA

## Introduction to the SGMII Implementation

The Serial GMII (SGMII) is an alternative interface to the GMII/MII that converts the parallel interface of the GMII/MII into a serial format that is capable of carrying traffic at speeds of 10 Mb/s, 100 Mb/s, and 1 Gb/s. It radically reduces the I/O count and is, therefore, often favored by PCB designers.

The SGMII physical interface was defined by Cisco Systems. The data signals operate at a rate of 1.25 Gb/s. Due to the speed of these signals, differential pairs are used to provide signal integrity and minimize noise.

The source-synchronous clock signals defined in the specification are not implemented in the Ethernet MAC. Instead, the serial transceiver is used to transmit and receive the differential data at the required rate using clock data recovery (CDR). For more information on SGMII, refer to the *Serial GMII specification v1.7*.

Figure 6-24 shows an expanded diagram of the PCS/PMA sublayer block. The Ethernet MAC is connected to a serial transceiver, which in turn connects to a Tri-Speed Ethernet BASE-T PHY device via SGMII.

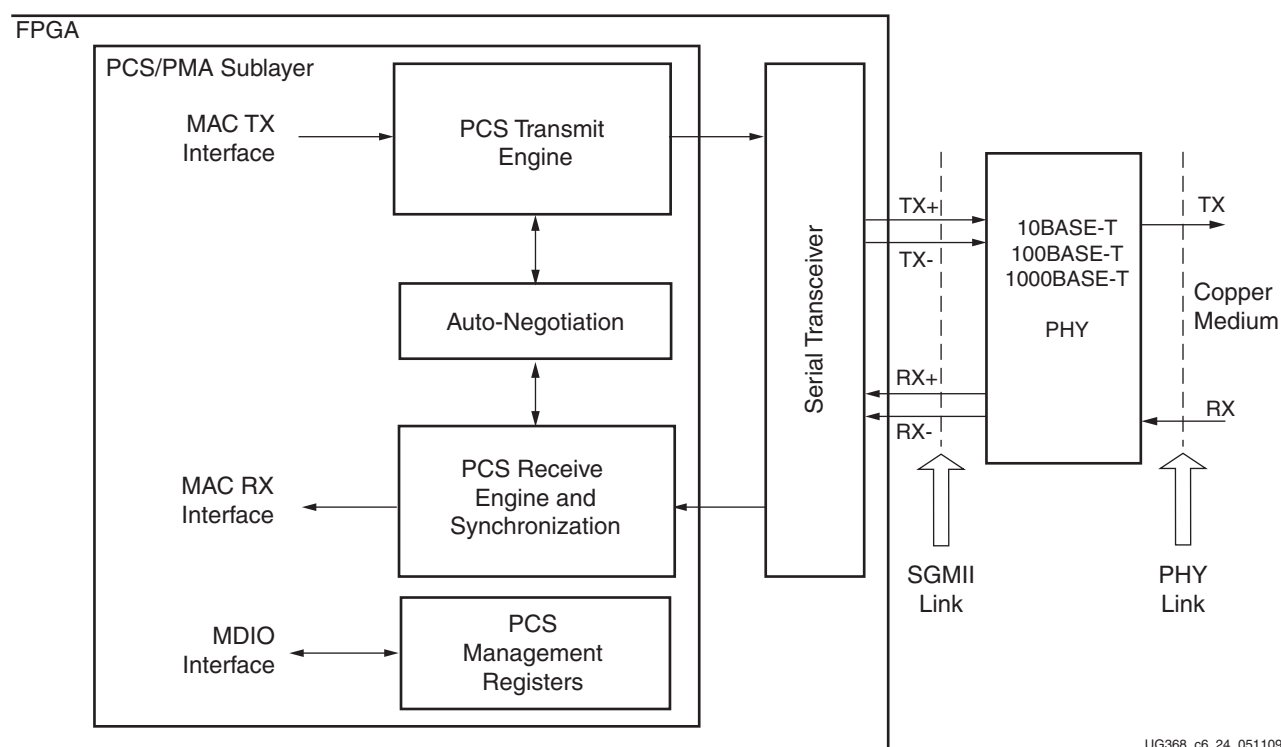


Figure 6-24: Ethernet PCS/PMA Sublayer Extension

The PCS/PMA sublayer contains its own functional blocks, which are illustrated and summarized in the following subsections.

## PCS Transmit Engine

The PCS Transmit Engine encodes the datastream received from the Ethernet MAC into a sequence of ordered sets. The data is then transmitted to the serial transceiver, using an 8-bit datapath, which then provides 8B/10B encoding of this data and parallel to serial conversion. The output from the serial transceiver is a differential pair operating at a rate of 1.25 Gb/s.

## PCS Receive Engine and Synchronization

The serial transceiver receives a serial differential pair operating at a rate of 1.25 Gb/s from the SGMII PHY. The serial transceiver, using CDR, performs word alignment on this serial datastream and then converts this to a parallel interface. This data is then 8B/10B decoded by the serial transceiver before presenting this to the PCS/PMA sublayer using an 8-bit datapath.

Within the PCS/PMA sublayer, the synchronization process performs analysis of valid/invalid received sequence ordered sets to determine if the link is in good order. The PCS receive engine then decodes these sequence ordered sets into a format that can then be presented to the standard receiver datapath within the Ethernet MAC.

## Auto-Negotiation

The SGMII auto-negotiation function allows the Ethernet MAC to communicate with the attached PHY device. Auto-negotiation is controlled and monitored using a software function through the PCS Management Registers. See [SGMII Auto-Negotiation, page 182](#).

## PCS Management Registers

Configuration and status of the PCS/PMA sublayer, including access to and from the auto-negotiation function, is via the PCS Management Registers. These registers are accessed through the serial MDIO, see [Chapter 5, MDIO Interface](#).

For the SGMII standard, the configuration and status registers available are listed in [SGMII Management Registers, page 125](#).

## SGMII RX Elastic Buffer

The RX elastic buffer can be implemented in one of two ways:

- Using the buffer present in the serial transceivers
- Using a larger buffer that is implemented in the FPGA logic

This section describes the selection and implementation of two options in the following subsections:

- [Serial Transceiver Logic Using the RX Elastic Buffer](#)
- [Serial Transceiver Logic Using the RX Elastic Buffer in FPGA Logic](#)

## RX Elastic Buffer Implementations

### Selecting the Buffer Implementation from the GUI

The GUI for the Ethernet MAC provides two options under the heading of SGMII Capabilities. These options are:

- Option 1: **10/100/1000 Mb/s clock tolerance compliant with Ethernet specification**
  - For this option, if the FPGA and PHY clocks are independent and each has 100 ppm tolerance, an additional elastic buffer is required (100 slices and 1 block RAM per Ethernet MAC).
- Option 2: **10/100/1000 Mb/s (restricted tolerance for clocks) OR 100/1000 Mb/s**
  - For this option, if 10 Mb/s is not required OR it is required but the FPGA and PHY clocks are closely related (see *Virtex-6 GTX Transceiver User Guide*), the RX buffer can be used (saves 100 slices and 1 block RAM per Ethernet MAC).

Option 1, the default mode, provides the implementation using the RX elastic buffer in the FPGA logic. This alternative RX elastic buffer utilizes a single block RAM to create a buffer that is twice as large as the one present in the serial transceiver, therefore consuming extra logic resources. However, this default mode provides reliable SGMII operation.

Option 2 uses the RX elastic buffer in the serial transceivers. This buffer, which is half the size of the buffer in Option 1, can potentially underflow or overflow during SGMII frame reception at 10 Mb/s operation (see [The FPGA RX Elastic Buffer Requirement](#)). However, in logical implementations where this case is proven reliable, this option is preferred because of its lower logic utilization.

### The FPGA RX Elastic Buffer Requirement

[Figure 6-25](#) illustrates a simplified diagram of a common situation where the Ethernet MAC in SGMII mode is interfaced to an external PHY device. Separate oscillator sources are used for the FPGA and the external PHY. IEEE Std 802.3-2005 uses clock sources with a 100 ppm tolerance. In [Figure 6-25](#), the clock source for the PHY is slightly faster than the clock source to the FPGA. Therefore, during frame reception, the RX elastic buffer (implemented in the serial transceiver in this example) starts to fill up.

Following frame reception in the interframe gap period, idles are removed from the received datastream to return the RX elastic buffer to half-full occupancy. This task is performed by the clock correction circuitry (see *Virtex-6 GTX Transceiver User Guide*).

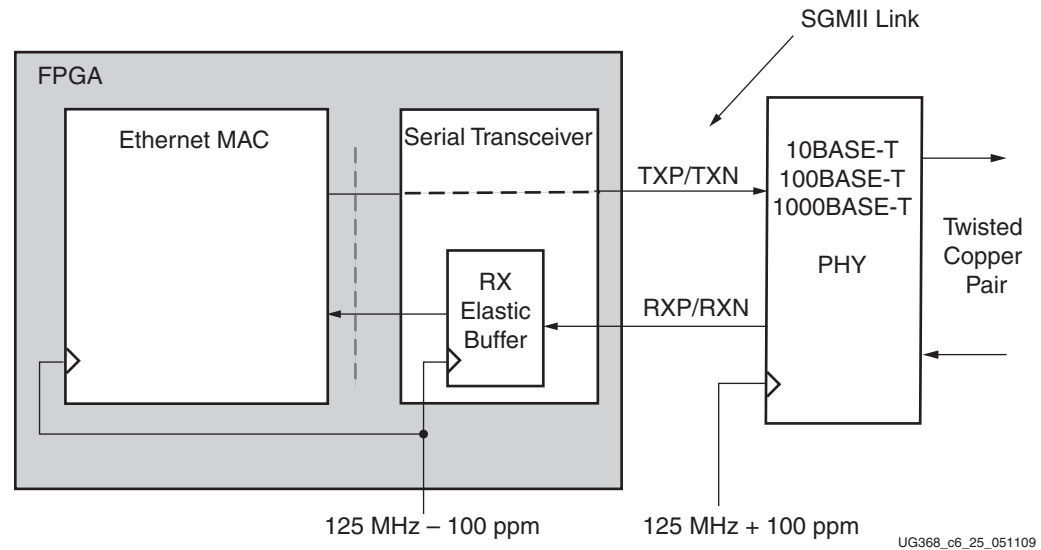


Figure 6-25: SGMII Implementation Using Separate Clock Sources

Assuming separate clock sources, each with 100 ppm tolerance, the maximum frequency difference between the two devices can be 200 ppm, which translates into a full clock period difference every 5000 clock periods.

Relating this information to an Ethernet frame, there is a single byte of difference every 5000 bytes of received frame data, causing the RX elastic buffer to either fill or empty by an occupancy of one.

The maximum Ethernet frame (non-jumbo) has a 1522-byte size for a VLAN frame:

- At 1 Gb/s operation, this size translates into 1522 clock cycles
- At 100 Mb/s operation, this size translates into 15220 clock cycles (because each byte is repeated 10 times)
- At 10 Mb/s operation, this size translates into 152200 clock cycles (because each byte is repeated 100 times)

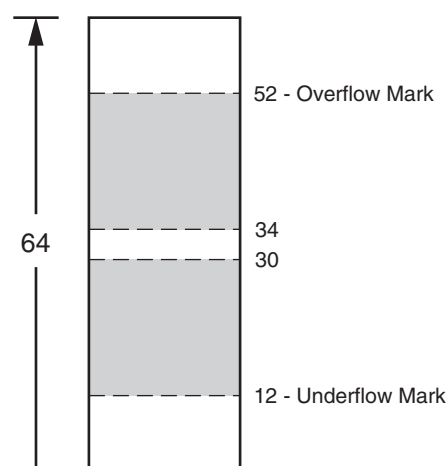
Considering the 10 Mb/s case,  $152200/5000 = 31$  FIFO entries are needed in the elastic buffer above and below the halfway point to ensure that the buffer does not underflow or overflow during frame reception. This assumes that frame reception begins when the buffer is exactly half full.

### Serial Transceiver Elastic Buffer

The RX elastic buffer in the serial transceivers has 64 entries. However, the buffer cannot be assumed to be exactly half full at the start of frame reception. Additionally, the underflow and overflow thresholds are not exact.

Figure 6-26 illustrates the elastic buffer depths and thresholds of serial transceivers in Virtex-6 devices. Each FIFO word corresponds to a single character of data (equivalent to a single byte of data following 8B/10B decoding).

Serial Transceiver RX Elastic Buffer



UG368\_c6\_26\_051109

**Figure 6-26: Elastic Buffer Size for Serial Transceivers**

The shaded area represents the usable buffer for the duration of frame reception.

- If the buffer is filling during frame reception, then  $52 - 34 = 18$  FIFO locations are available before the buffer hits the overflow mark.
- If the buffer is emptying during reception, then  $30 - 12 = 18$  FIFO locations are available before the buffer hits the underflow mark.

This analysis assumes that the buffer is approximately at the half-full level at the start of the frame reception. There are, as illustrated, two locations of uncertainty above and below the exact half-full mark of 32. The uncertainty is a result of the clock correction decision, which is performed in a different clock domain.

Since there is a worst case scenario of one clock edge difference every 5,000 clock periods, the maximum number of clock cycles (bytes) that can exist in a single frame passing through the buffer before an error occurs is  $5,000 \times 18 = 90,000$  bytes.

Table 6-2 translates this into maximum frame lengths at different Ethernet MAC speeds. The situation is worse at SGMII speeds lower than 1 Gb/s because bytes are repeated multiple times.

**Table 6-2: Maximum Frame Sizes for Serial Transceiver RX Elastic Buffers (100 ppm Clock Tolerance)**

Standard	Speed	Maximum Frame Size
SGMII	1 Gb/s	90,000
SGMII	100 Mb/s	9,000
SGMII	10 Mb/s	900

## FPGA Logic Elastic Buffer

To achieve reliable SGMII operation at 10 Mb/s (non-jumbo frames), the example design provides an RX elastic buffer implemented in the FPGA fabric. It is twice the size of the serial transceiver's RX elastic buffer, and nominally provides 64 entries above and below the half-full threshold. This configuration can manage standard (non-jumbo) Ethernet frames at all three SGMII speeds.

Figure 6-27 illustrates alternative FPGA logic RX elastic buffer depth and thresholds. Each FIFO word corresponds to a single character of data (equivalent to a single byte of data following 8B/10B decoding). This buffer can optionally be used to replace the RX elastic buffers of the serial transceiver. See [Serial Transceiver Logic Using the RX Elastic Buffer in FPGA Logic](#), page 178.

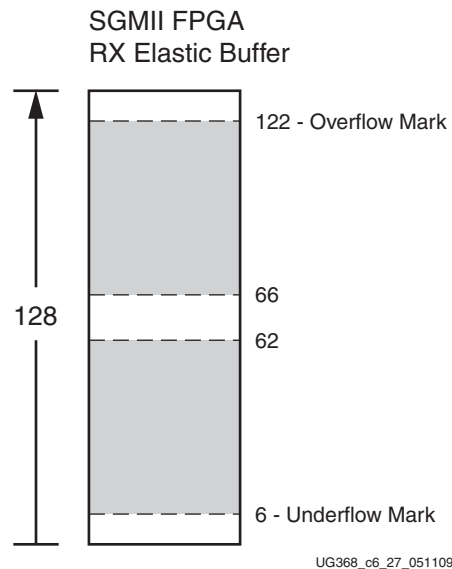


Figure 6-27: Elastic Buffer Size for FPGA Logic Buffer

The shaded area in Figure 6-27 represents the usable buffer availability for the duration of frame reception.

- If the buffer is filling during frame reception, then  $122 - 66 = 56$  FIFO locations are available before the buffer hits the overflow mark.
- If the buffer is emptying during reception, then  $62 - 6 = 56$  FIFO locations are available before the buffer hits the underflow mark.

This analysis assumes that the buffer is approximately at the half-full level at the start of the frame reception. As illustrated in Figure 6-27, there are two locations of uncertainty above and below the exact half-full mark of 64 as a result of the clock correction decision, which is based across an asynchronous boundary.

Since there is a worst-case scenario of one clock edge difference every 5,000 clock periods, the maximum number of clock cycles (bytes) that can exist in a single frame passing through the buffer before an error occurs is  $5,000 \times 56 = 280,000$  bytes.

Table 6-3 translates this into maximum frame lengths at different Ethernet MAC speeds. The situation is worse at SGMII speeds lower than 1 Gb/s because bytes are repeated multiple times.

**Table 6-3: Maximum Frame Sizes for Serial Transceiver RX Elastic Buffers (100 ppm Clock Tolerance)**

Standard	Speed	Maximum Frame Size
SGMII	1 Gb/s	280,000
SGMII	100 Mb/s	28,000
SGMII	10 Mb/s	2,800

## Using the Serial Transceiver RX Elastic Buffer

The RX elastic buffer implemented in the serial transceiver can be used reliably when:

- 10 Mb/s operation is not required. Both 1 Gb/s and 100 Mb/s operate on standard Ethernet MAC frame sizes only.
- When the clocks are closely related (see [Closely Related Clock Sources](#)).

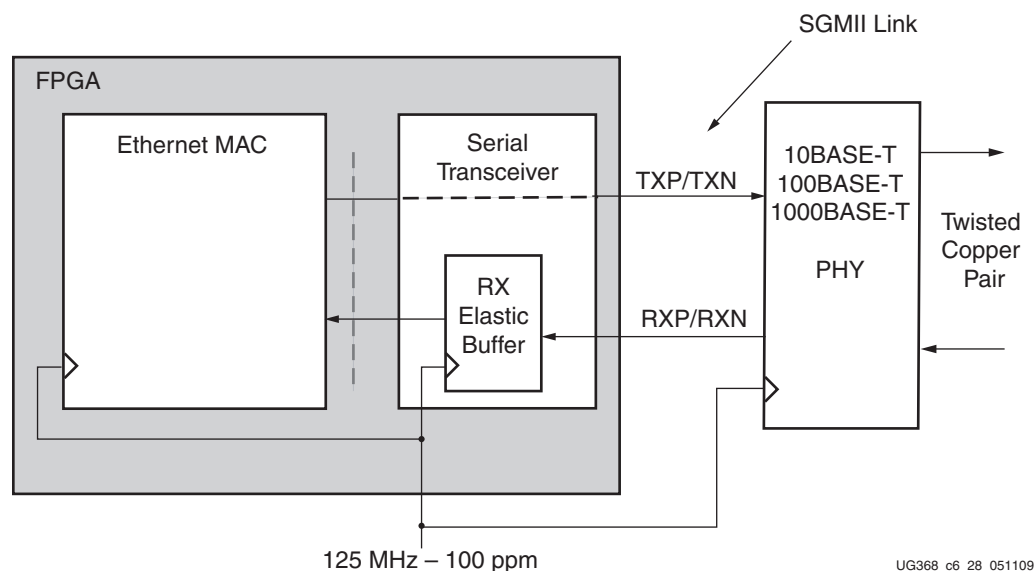
Designers are recommended to select the FPGA Logic RX elastic buffer implementation if they have any doubts to reliable operation.

## Closely Related Clock Sources

Two cases are described with closely related clocks in SGMII mode:

- Case 1

[Figure 6-28](#) illustrates a simplified diagram of a common situation where the Ethernet MAC in SGMII mode is interfaced to an external PHY device. A common oscillator source is used for both the FPGA and the external PHY.



**Figure 6-28: SGMII Implementation Using Shared Clock Sources**

If the PHY device sources the receiver SGMII stream synchronously from the shared oscillator (refer to the PHY data sheet), the serial transceiver receives data at exactly the same rate as that used by the core. That is, the RX elastic buffer neither empties nor fills because the same frequency clock is on either side.



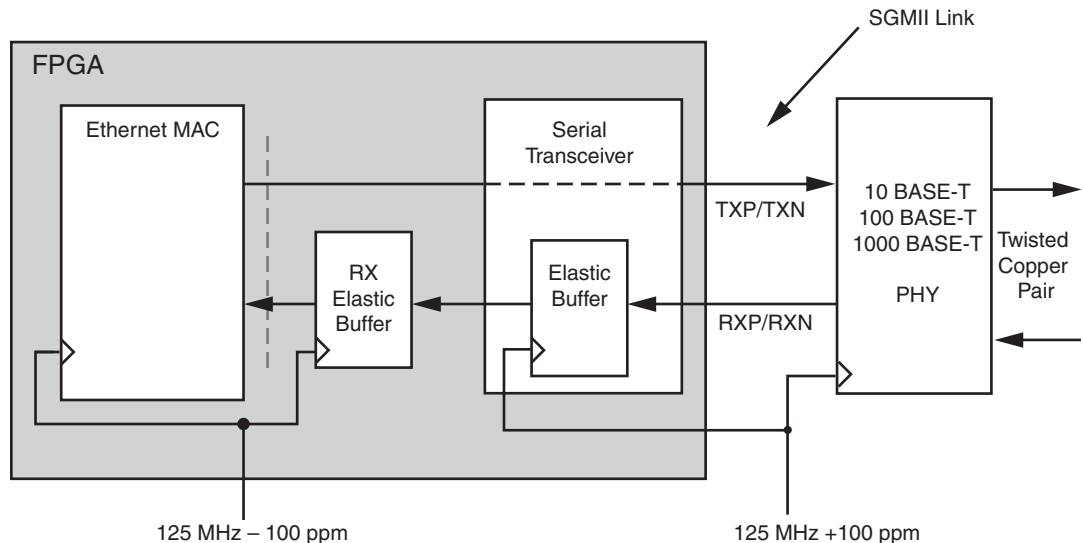
In this situation, the RX elastic buffer does not underflow or overflow, and the RX elastic buffer implementation in the serial transceiver is recommended to save logic resources.

- Case 2

Using the case illustrated by Figure 6-25, assume that both clock sources used are 50 ppm. The maximum frequency difference between the two devices is 100 ppm, translating into a full clock period difference every 10,000 clock periods and resulting in a requirement for 16 FIFO entries above and below the half-full point. This case provides reliable operation with the serial transceiver RX elastic buffers. However, the designer must check the PHY data sheet to ensure that the PHY device sources the receiver SGMII stream synchronously to its reference oscillator.

### Using the FPGA Logic Elastic Buffer

Figure 6-29 illustrates a simplified diagram of a situation where the Ethernet MAC in SGMII mode is interfaced to an external PHY device with an independent clock. The larger FPGA elastic buffer is used to cross clock domains, rather than the serial transceiver's elastic buffer.



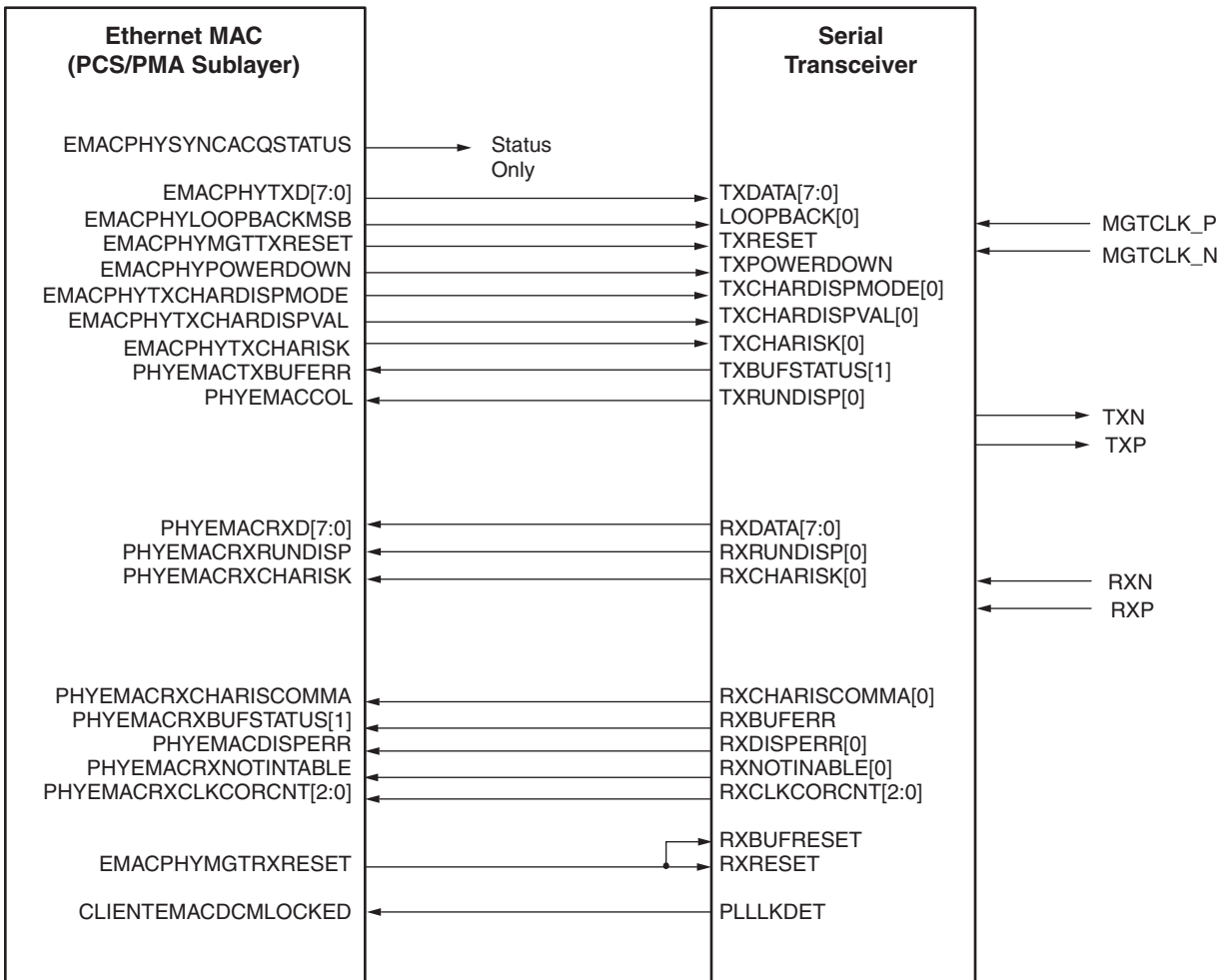
UG368\_c6\_29\_051109

Figure 6-29: SGMII Implementation Using a Logic Buffer

Using the SGMII in this configuration eliminates the possibility of buffer error if the clocks are not tightly controlled enough to use the serial transceiver elastic buffer.

### Serial Transceiver Logic Using the RX Elastic Buffer

Figure 6-30 shows the Ethernet MAC configured with SGMII as the physical interface. Connections to the Virtex-6 FPGA serial transceiver are illustrated.



UG368\_6\_30\_051109

**Figure 6-30: Ethernet MAC Configured in SGMII Mode**

These connections and the shim are created by the CORE Generator tool when the physical interface is selected to be either SGMII or 1000BASE-X PCS/PMA. By using the CORE Generator tool, the time required to instantiate the Ethernet MAC into a usable design is greatly reduced. See [Accessing the Ethernet MAC from the CORE Generator Tool](#), page 21.

## Serial Transceiver Logic Using the RX Elastic Buffer in FPGA Logic

The example design, delivered with the core in the CORE Generator tool, is split between two different hierarchical layers. The block level is designed so that it can be instantiated directly into customer designs. It provides the following functionality:

- Instantiates the core from HDL
- Connects the physical-side interface of the core to a Virtex-6 FPGA serial transceiver via the RX elastic buffer in FPGA logic

The Ethernet MAC is designed to integrate with the Virtex-6 FPGA serial transceiver. The connections and logic required between the Ethernet MAC and serial transceiver are illustrated in [Figure 6-31](#). The signal names and logic shown in [Figure 6-31](#) exactly match those delivered with the example design when the serial transceiver is used.

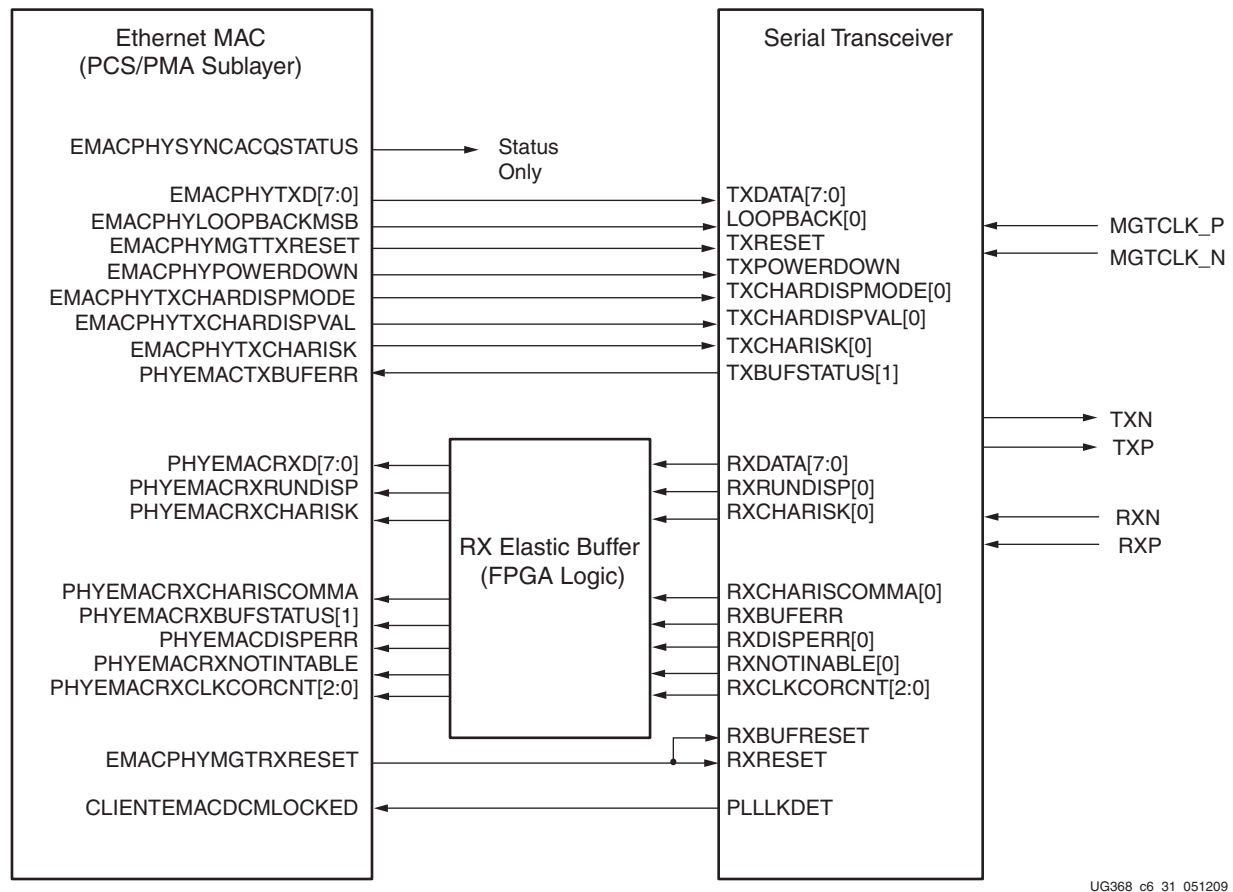


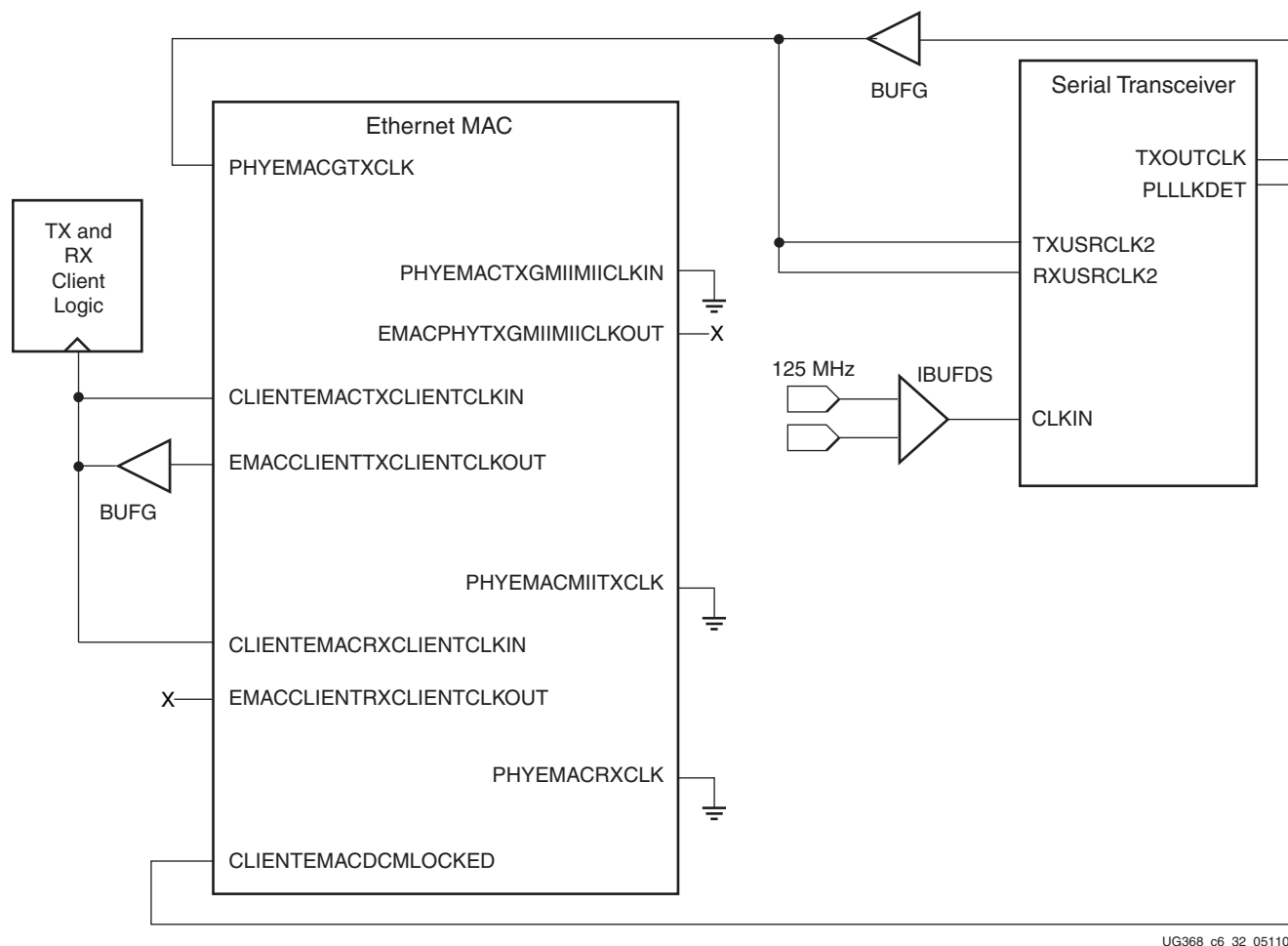
Figure 6-31: SGMII Connection to a Virtex-6 FPGA Serial Transceiver

Figure 6-31 shows that the RX elastic buffer is implemented in the FPGA logic between the serial transceiver and the Ethernet MAC instead of in the serial transceiver. This alternative RX elastic buffer utilizes a single block RAM to create a buffer that is twice as large as the one present in the serial transceiver. This configuration, therefore, can handle larger frame sizes before clock tolerances accumulate and result in emptying or filling of the buffer. This is necessary for SGMII operation at 10 Mb/s, where each frame size is effectively 100 times larger than the same frame is at 1 Gb/s due to each byte being repeated 100 times.

## SGMII Clock Management

### Tri-Speed Operation

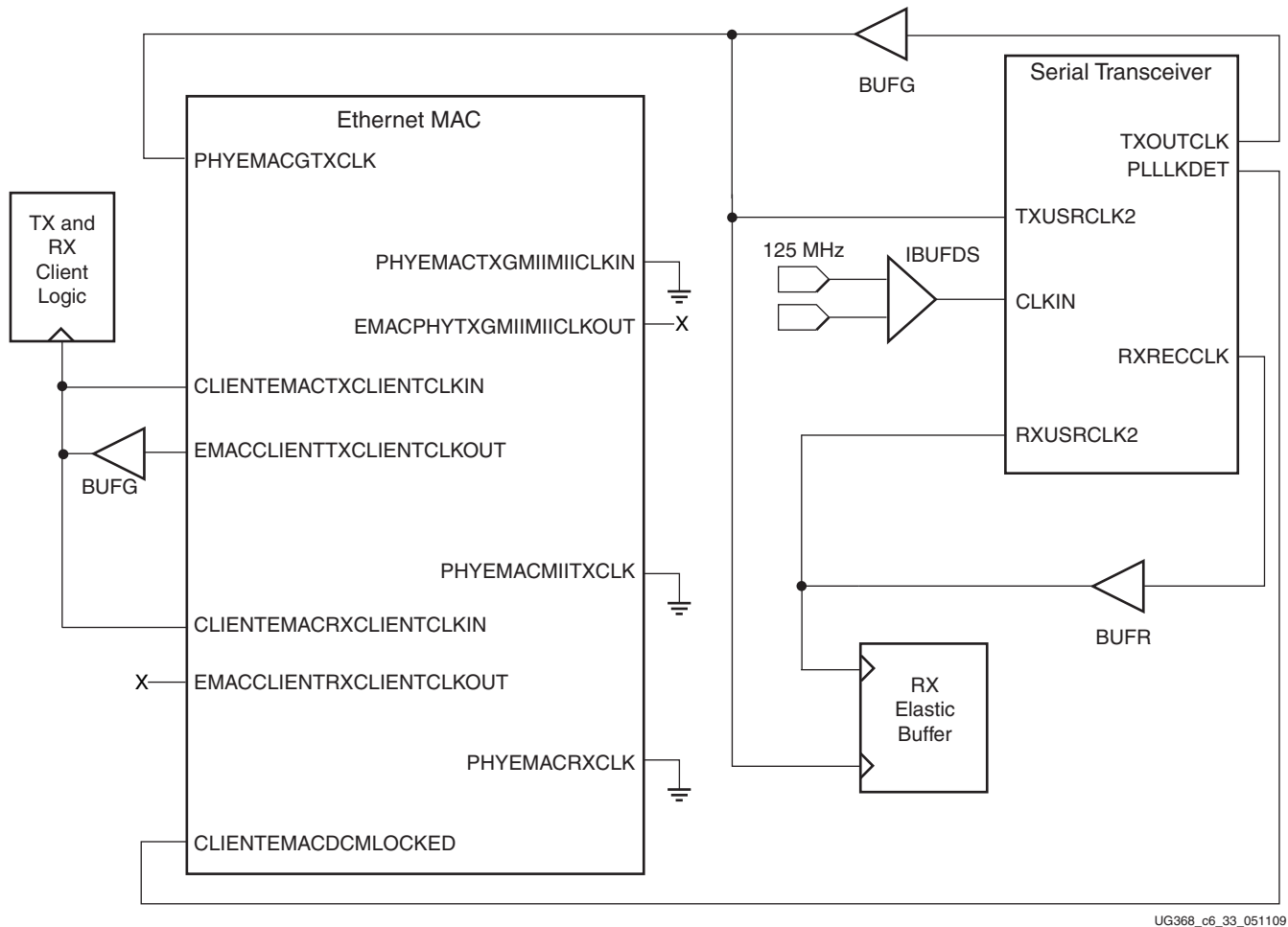
Figure 6-32 shows the clock management used with the SGMII interface without the FPGA logic elastic buffer. The CLKIN inputs to the serial transceiver must be connected to an external, high-quality differential reference clock with frequency of 125 MHz. A 125 MHz clock source is then provided to the FPGA logic from the TXOUTCLK output port of the serial transceiver. This is connected to global clock routing using a BUFG as illustrated. This clock should then be routed to the PHYEMACGTCLK of the Ethernet MAC.



UG368\_c6\_32\_051109

Figure 6-32: SGMII Clock Management - Serial Transceiver RX Elastic Buffer

When using the larger FPGA fabric implementation of the RX elastic buffer, data is clocked out of the serial transceiver synchronously to RXRECCLK. This clock, placed on a BUFR component, drives RXUSRCLK2 and some of the logic in the RX elastic buffer, as illustrated in Figure 6-33.



UG368\_c6\_33\_051109

Figure 6-33: Using the FPGA Fabric RX Elastic Buffer

In both configurations, the PLLKDET signal from the serial transceiver (indicating that its internal PLLs have locked) is routed to the CLIENTEMACDCMLOCKED input port of the Ethernet MAC, which ensures that the state machines of the Ethernet MAC are held in reset until the serial transceiver has locked and its clocks are running cleanly.

Either of the EMACCLIENTRXCLIENTCLKOUT or EMACCLIENTTXCLIENTCLKOUT output ports can be used to obtain the clock used for the Ethernet MAC client logic (both receiver and transmitter logic can share the same clock) with the unused clock being left unconnected. EMACCLIENTTXCLIENTCLKOUT is used in our example and is connected to a BUFG, which then provides the client clock to FPGA logic. It must also be routed back to the Ethernet MAC through the CLIENTEMACRXCLIENTCLKIN and CLIENTEMACTXCLIENTCLKIN input ports.

As described in [Appendix B, Ethernet MAC Clocks](#), the following clock signals are unused:

- EMACPHYTXGMIIMIICLKOUT
- EMACCLIENTRXCLIENTCLKOUT
- PHYEMACTXGMIIMIICLKIN
- PHYEMACMIITXCLK
- PHYEMACRXCLK

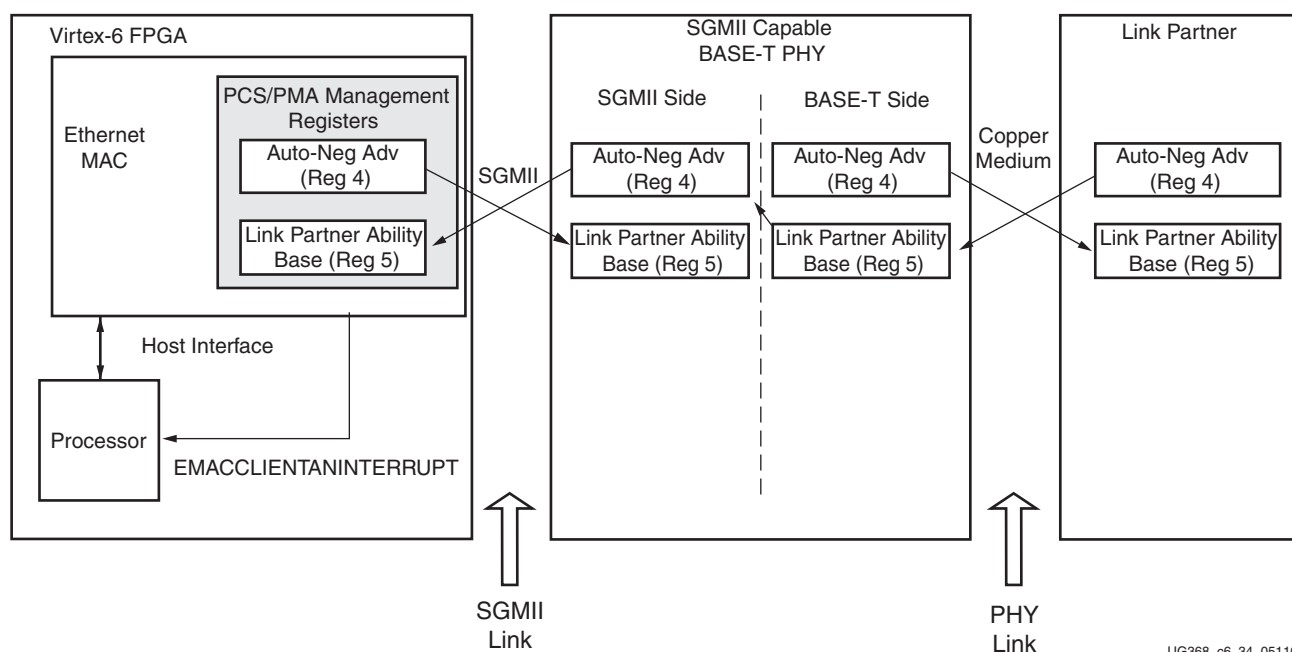
## 1 Gb/s Only Operation

When the SGMII is used only for 1 Gb/s speeds, further clocking optimization can be performed. The clock logic is then identical to that described in [1000BASE-X PCS/PMA Clock Management](#), page 161.

## SGMII Auto-Negotiation

### Overview of Operation

[Figure 6-34](#) illustrates a simplified diagram of the Ethernet MAC instantiated within a Virtex-6 device. The only components shown are two of the PCS Management Registers, which are directly involved in the auto-negotiation process (see [SGMII Management Registers](#), page 125). The corresponding registers of the connected devices are also shown.



UG368\_c6\_34\_051109

Figure 6-34: SGMII Auto-Negotiation Overview

The SGMII-capable PHY has two distinctive sides to auto-negotiation as illustrated:

- The PHY initially performs auto-negotiation with its link partner across the PHY link using the relevant auto-negotiation standard for the chosen medium (BASE-T auto-negotiation, illustrated in [Figure 6-34](#), uses a copper medium). This resolves the operational speed and duplex mode with the link partner.
- The PHY then initiates a secondary auto-negotiation process with the Ethernet MAC across the SGMII link. This leverages the 1000BASE-X auto-negotiation specification described in [1000BASE-X Auto-Negotiation](#) with only minor differences. This transfers the results of the initial PHY with link partner auto-negotiation across the SGMII, and this is the only auto-negotiation process observed by the Ethernet MAC.

The SGMII auto-negotiation function leverages the [1000BASE-X Auto-Negotiation](#) function with the exception of:

- The duration of the link timer of the SGMII auto-negotiation decreases from 10 ms to 1.6 ms making the entire auto-negotiation cycle faster (see [SGMII Auto-Negotiation Link Timer](#)).
- The information exchanged now contains speed resolution in addition to duplex mode.

Under normal conditions, this completes the auto-negotiation information exchange. The results can be read from the [Table 5-19, page 128](#). The duplex mode and speed of the Ethernet MAC should then be configured by a software routine to match. This does not happen automatically within the Ethernet MAC. There are two methods by which a host processor can learn of the completion of an auto-negotiation cycle:

- By polling the auto-negotiation completion bit 1.5 in [Table 5-16, page 127](#).
- By using the auto-negotiation interrupt port (see [Auto-Negotiation Interrupt, page 184](#)).

## SGMII Auto-Negotiation Link Timer

The auto-negotiation link timer is used to time three phases of the auto-negotiation procedure. For the SGMII standard, this link timer is defined as having a duration of 1.6 ms.

The duration of the link timer used by the Ethernet MAC can be configured with the EMAC\_LINKTIMERVAL[8:0] attribute (see [Table 2-18, page 41](#)) or by writing to the appropriate host register (see [Table 4-6, page 86](#)). The duration of the timer is approximately equal to the binary value placed onto this attribute multiplied by 32.768  $\mu$ s (4096 clock periods of the 125 MHz clock provided to the Ethernet MAC on PHYEMACGTCLK). The accuracy of this link timer is within the range:

+0 to -32.768  $\mu$ s

Therefore, for the SGMII standard, set the EMAC\_LINKTIMERVAL[8:0] attribute to:

000110010 = 50 decimal

This corresponds to a timer duration of between 1.606 and 1.638 ms. This value can be reduced for simulation.

## Auto-Negotiation Interrupt

The auto-negotiation function has an EMACCLIENTANINTERRUPT port. This port is designed to be used with common microprocessor bus architectures.

The operation of this port is enabled or disabled and cleared via the [SGMII Management Registers, page 125](#).

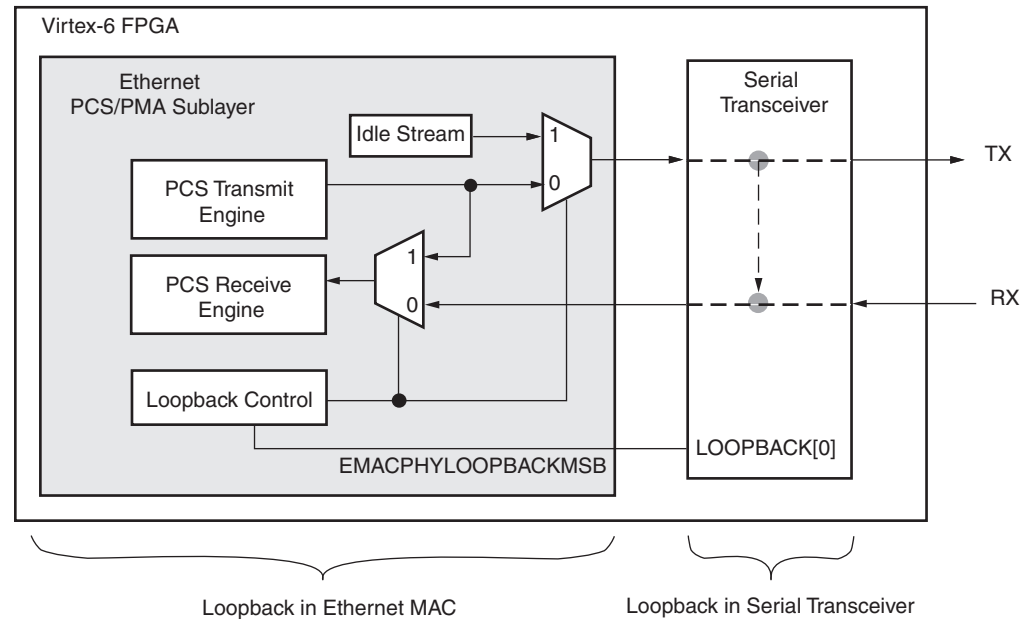
- When disabled, this port is permanently driven Low.
- When enabled, this port is set to logic 1 following the completion of an auto-negotiation cycle. It remains High until cleared after a zero is written to bit 16.1 (Interrupt Status bit) of the [SGMII Management Registers](#).

## Loopback When Using the PCS/PMA Sublayer for SGMII

[Figure 6-35](#) illustrates the loopback options when using the PCS/PMA sublayer. Two possible loopback positions are illustrated:

- **Loopback in the Ethernet MAC.** When placed into loopback, data is routed from the transmitter to the receiver path at the last possible point in the PCS/PMA sublayer, immediately before the serial transceiver interface. When placed into loopback, a constant stream of Idle code groups are transmitted through the serial transceiver. Loopback in this position allows test frames to be looped back within the Ethernet MAC without allowing them to be received by the link partner. The transmission of idles allows the link partner to remain in synchronization so that no fault is reported.
- **Loopback in the serial transceiver.** The serial transceiver can alternatively be switched into loopback by connecting the EMACPHYLOOPBACKMSB port of the Ethernet MAC to the loopback port of the serial transceiver as illustrated. The serial transceiver loopback routes data from the transmitter path to the receiver path within the serial transceiver. However, this data is also transmitted out of the serial transceiver, so any test frames used for a loopback test are received by the link partner.





UG368\_c6\_35\_0051109

Figure 6-35: Loopback When Using the PCS/PMA Sublayer

## Switching Between SGMII and 1000BASE-X Standards

When the Ethernet MAC is configured for tri-speed operation with the SGMII physical interface (see [SGMII Clock Management](#)), it is possible to switch between SGMII and 1000BASE-X modes without reconfiguring the FPGA. This includes transitions from SGMII to 1000BASE-X and from 1000BASE-X to SGMII.

Mode switching is useful when using off-the-shelf PHY devices with the ability to perform both BASE-X and BASE-T standards, or when a single Virtex-6 FPGA bitstream can be used with either a BASE-X or BASE-T PHY device.

The default method for switching modes follows. This task can be simplified in some cases by using the SGMII/1000BASE-X mode switching block provided by the CORE Generator tool, which automates the tasks shown in step 1, below. See [Accessing the Ethernet MAC from the CORE Generator Tool](#) for more details on using the CORE Generator tool.

## Performing a Mode Switch

The Ethernet MAC mode of operation is set by writing to the Ethernet MAC Mode Configuration Register via the host interface (see [Table 4-6, page 86](#)). [Table 6-4](#) indicates the possible settings.

**Table 6-4: Possible Mode Switch Settings**

Mode of Operation	Bit 28	Bit 27
Illegal setting	0	0
1000BASE-X	0	1
SGMII	1	0
Illegal setting	1	1

After the appropriate Ethernet MAC Mode Configuration Register bits have been written, the mode switch begins to take effect immediately.

The programmable link timer value is also set via the Ethernet MAC Mode Configuration Register, and it is recommended that the link timer value appropriate to the new mode of operation is written at the same time as the mode switch command. See [1000BASE-X PCS/PMA Management Registers, page 118](#).

The PCS/PMA Management Registers accessible via MDIO have different interpretations for the two standards. Following a mode switch, these registers should immediately be interpreted under the new standard of operation.

- For SGMII operation, PCS/PMA Management Registers should be interpreted according to the format in [SGMII Management Registers, page 125](#).
- For 1000BASE-X operation, PCS/PMA Management Registers should be interpreted according to the format in [1000BASE-X PCS/PMA, page 158](#).

When switching from SGMII to 1000BASE-X, the auto-negotiation advertisement register (Register 4) must be reprogrammed to the desired settings.

After the mode switch command and the new link timer value have been written, and the auto-negotiation advertisement register has been reprogrammed (if appropriate), the user must request an auto-negotiation restart by setting bit 0.9 in the appropriate control register. See [Table 5-3, page 119](#) and [Table 5-15, page 126](#) for details.

The mode switch process is finished following the successful completion of auto-negotiation. Other requirements and considerations for mode switching follow.

## Requirements and Considerations for Mode Switching

To support switching between SGMII and 1000BASE-X standards, the Ethernet MAC must be configured for tri-speed operation with the SGMII physical interface. This configuration properly supports the 10/100/1000 Mb/s operation of the SGMII standard, in addition to supporting the 1000 Mb/s operation of the 1000BASE-X standard.

Mode switching is only allowed following initial power-up or after the deassertion of reset.

The Ethernet MAC must operate in 1000BASE-X mode to use an optical fiber medium. It must operate in SGMII mode to provide BASE-T functionality and use a twisted-pair copper medium.

## Interfacing to a Statistics Block

To collect statistics information from an individual Ethernet MAC, a custom statistics counter can be implemented in the FPGA logic. A parameterizable Ethernet Statistics LogiCORE™ solution that fulfills all the requirements for statistics information collection is available through the CORE Generator™ tool ([DS323](#), *LogiCORE Ethernet Statistics Data Sheet*, which provides a full description of the Ethernet Statistics LogiCORE™ block). Each Ethernet MAC that is used requires its own instance of this Statistics IP, as shown in [Figure 2-1](#), [page 23](#).

This chapter describes how to connect the Ethernet MAC block to a Statistics block to allow the statistics registers of each Ethernet MAC to be accessed by either the host or DCR bus interface. A description of the address space that can be used for the statistics registers is provided to avoid register access contentions.

[Statistics Vectors](#), [page 74](#) provides details on the statistics interface of the Ethernet MAC and how the output signals encode the statistics vector information.

### Using the Host Bus to Access Statistics Registers

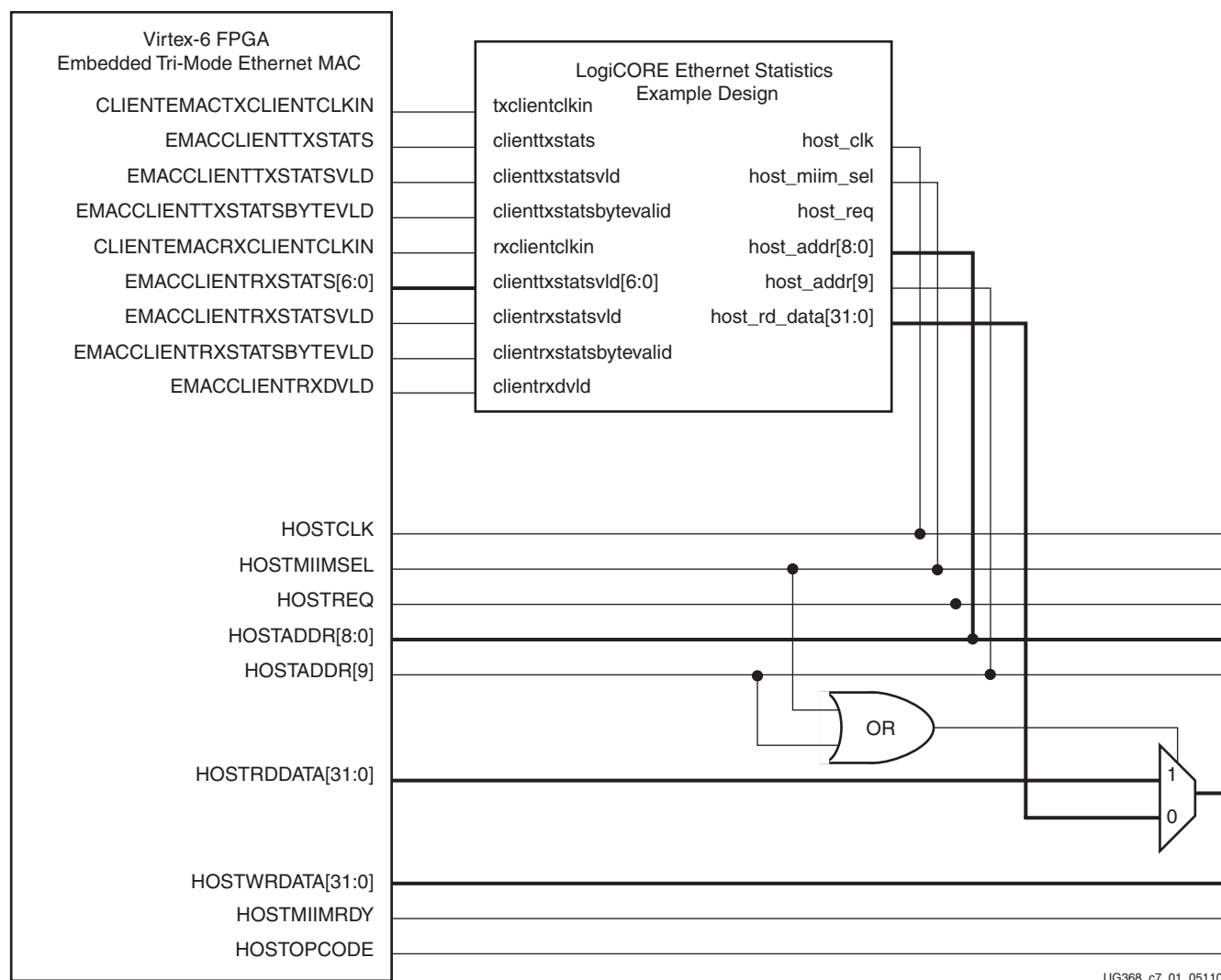
When the Ethernet MAC is used with the host bus, interfacing to an FPGA logic-based statistics block is straight-forward. The statistics values can be read via a host interface that is shared between the statistics counters and the Ethernet MAC block.

To share the host bus without contention, statistics counters need to use a different address space than the Ethernet MAC configuration registers. Conflicts with MDIO register accesses are avoided by only accessing statistics counters when the HOSTMIIMSEL signal is at logic 0. Implementation of the addressing scheme shown in [Table 7-1](#) ensures that the host bus can be shared without contention. This scheme provides space to address 512 statistics counters per Ethernet MAC, using addresses 0x000 to 0x1FF.

**Table 7-1: Addressing Scheme**

Transaction	Host_miim_sel	Host_addr[9]
Configuration	0	1
MIIM Access	1	X
Statistics Access	0	0

[Figure 7-1](#) shows how to integrate the Ethernet MAC with the LogiCORE Ethernet Statistics block, where the Ethernet statistics counters are accessed via the host bus.



UG368\_c7\_01\_051109

Figure 7-1: Host Bus to Ethernet Statistics Connection

## Using the DCR Bus to Access Statistics Registers

When the DCR bus interface of the Ethernet MAC is controlled by an embedded processor, the host interface and DCR bridge of the Ethernet MAC allow DCR bus accesses to control the unused host bus pins. The Ethernet MAC host bus interface can then access the statistics counters in the FPGA logic, controlled by register accesses on the DCR bus.

The host bus I/O signals of the Ethernet MAC are enabled for statistics counter access when a DCR read operation is made to address codes 0x000 to 0x02F and 0x040 to 0x04F inclusive (Table 4-15, page 97 describes the DCR address code space). This use of the host bus I/O signals provides a means of accessing the FPGA logic from the processor with space for 64 addresses.

When the DCR bus is instructed to access registers in this address code region, the DCR bridge translates the DCR commands into generic host read signals on the host bus I/O signals. The DCR transaction is encoded on the host bus signals HOSTRDDATA[31:0] and HOSTMIIMRDY as described in Table 4-23, page 108 and Figure 4-10. These signals can

access statistics counters in the same way as if a stand-alone host bus is used. The statistics values read from statistics counters are captured from the host bus signals HOSTWRDATA[31:0] as shown in Figure 4-10. The data read from the host bus can then be accessed by reading the DCR data registers.

Figure 7-2 shows how to integrate the Ethernet MAC with the LogiCORE Ethernet Statistics block, where the LogiCORE Ethernet statistics counters are accessed via the DCR bus. DS323, *LogiCORE Ethernet Statistics Data Sheet*, provides a full description of the Ethernet Statistics LogiCORE block.

An example of how to read from the statistics counters through the DCR bus is provided in the example in [Accessing FPGA Logic via Unused Host Bus Pins](#), page 107.

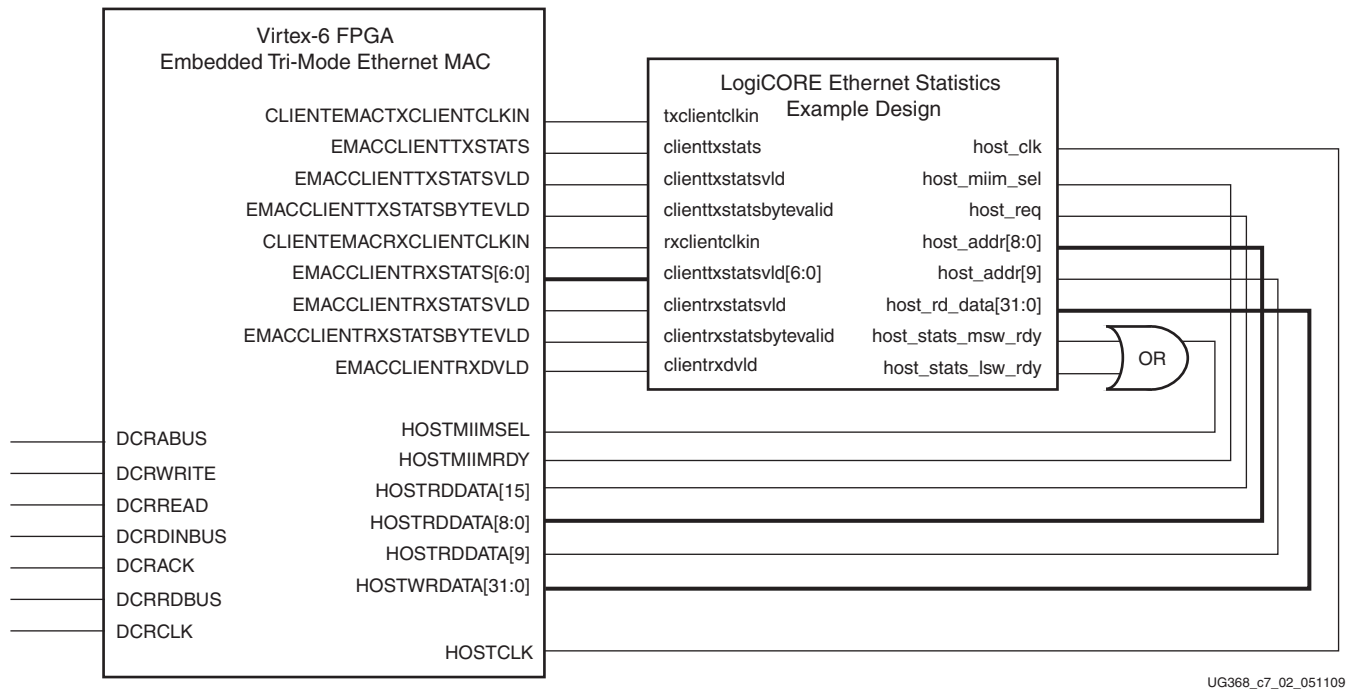


Figure 7-2: DCR Bus to Ethernet Statistics Connection



## Pinout Requirements

---

Xilinx recommends the following guidelines to improve design timing using the Virtex®-6 FPGA Embedded Tri-Mode Ethernet MAC:

- If available, use dedicated global clock pins for the Ethernet MAC input clocks.
- Use the column of IOBs located closest to the Ethernet MAC block.
- For 1000BASE-X PCS/PMA or SGMII physical interface configurations, use the serial transceiver located closest to the Ethernet MAC block.

## Regional Clocking Requirements

When implementing the Ethernet MAC with either a GMII or RGMII physical interface, regional clocking methodologies are used. For diagrams of specific clocking networks, refer to the appropriate configurations in [Chapter 6, Physical Interface](#).

The regional clocking networks, which utilize BUFIO and BUFR clock buffers in conjunction with IODELAY resources, substantially improve timing and are often necessary to achieve timing closure on the physical interface. However, they also impose certain pinout requirements which are critical to the planning process.

Specifically, the following three requirements must be met in order to support the regional clocking methodologies specified for GMII and RGMII physical interface implementations:

- The receive-side physical interface clock (GMII\_RX\_CLK for GMII, or RGMII\_RXC for RGMII) must be placed at a clock-capable I/O (CCIO) pin.
- All receive-side physical interface signals (see [Table A-1](#)) must be placed at package pins that correspond to the same clock region as the receive-side physical interface clock.
- An available Embedded Tri-Mode Ethernet MAC block must be present in a clock region that is reachable by the regionally buffered physical interface clock net.

The ideal solution is to place all receive-side physical interface signals within a single clock region, which also contains an Embedded Tri-Mode Ethernet MAC block. The receive-side physical interface clock must be placed at a CCIO in that region.

Other solutions are possible. When configuring the Ethernet MAC with a GMII or RGMII physical interface, it is highly recommended to study regional clocking capabilities and requirements during the pinout planning process. See the Virtex-6 FPGA User Guide for details.

**Table A-1: Receive-side Physical Interface Signals with Regional Clocking Requirements**

<b>GMII</b>	<b>RGMII</b>
GMII_RX_CLK (to CCIO pin)	RGMII_RXC (to CCIO pin)
GMII_RXD[7:0]	RGMII_RXD[3:0]
GMII_RX_DV	RGMII_RX_CTL
GMII_RX_ER	
GMII_COL	
GMII_CRS	



# *Ethernet MAC Clocks*

---

This appendix provides an overview of the Ethernet MAC clocking schemes. The clocking schemes that are internal to the Ethernet MAC block are introduced, followed by the clock connections to and from the FPGA logic. Finally, all clock input and output frequencies and definitions are listed for reference, dependent on the mode of operation.

The CORE Generator™ tool provides the appropriate clocking resources and configuration, greatly reducing the time required to properly integrate the Ethernet MAC into a design. See [Accessing the Ethernet MAC from the CORE Generator Tool](#), page 21.

## **Ethernet MAC Internal Clock Logic Overview**

The large number of clock signals can give an overwhelming first impression. [Figure B-1](#) is a simplified diagram to illustrate all Ethernet MAC input and output clock signals.

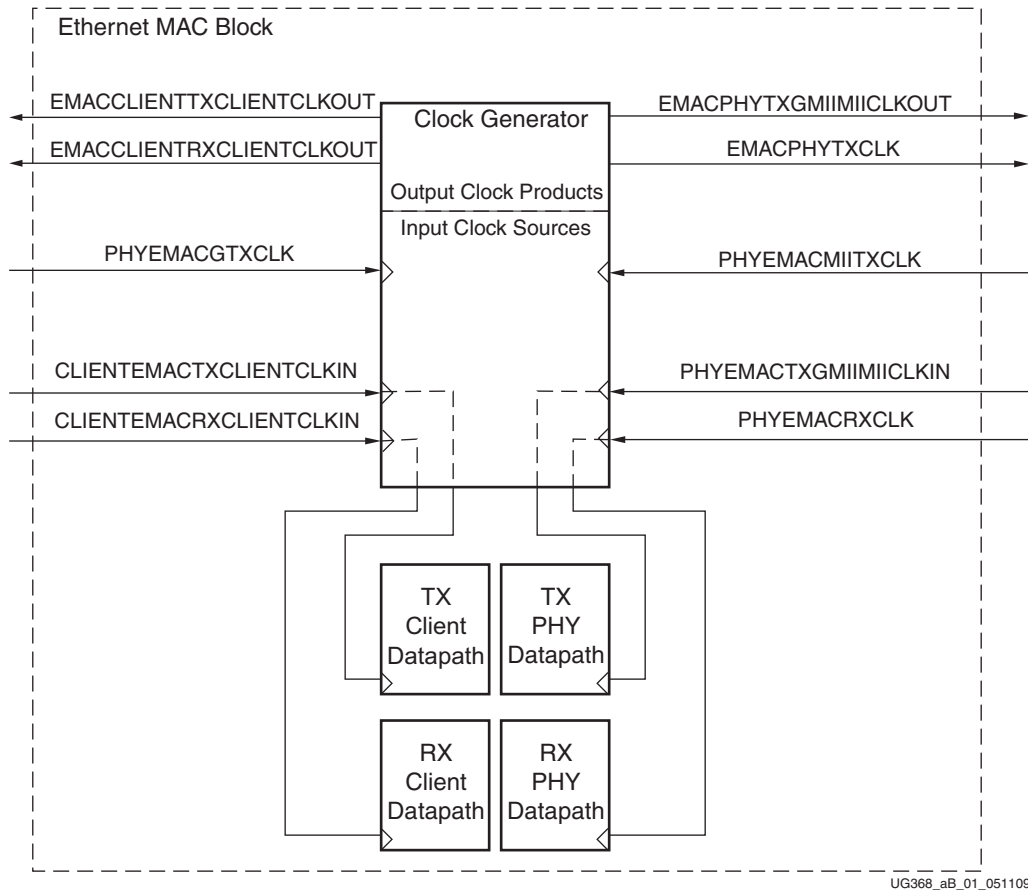


Figure B-1: Overview of Ethernet MAC Clock Circuitry

## Ethernet MAC Clock Generation

As shown in [Figure B-1](#), included in the Ethernet MAC is a Clock Generator module. This module is provided with input clock sources, from which the following output clock products are generated:

- EMACPHYTXGMIIMIICKOUT
- EMACPHYTXCLK
- EMACCLIENTTXCLIENTCLKOUT
- EMACCLIENTRXCLIENTCLKOUT

For these generated output clocks (refer to [Clock Definitions and Frequencies](#)):

- the output clocks are always at the correct frequency for their associated interfaces for any mode of operation and switch frequencies cleanly during speed changes.
- the output clocks are NOT used internally by any Ethernet MAC logic.
- the output clocks are provided only for the convenience of the user; they do not have to be used by the FPGA logic.

## Ethernet MAC Input Clocks

As shown in [Figure B-1](#), the following clock signals are input into the Ethernet MAC clock generator:

- PHYEMACGTCLK
- PHYEMACMIITXCLK
- PHYEMACRXCLK
- PHYEMACTXGMIIMICLKIN
- CLIENTEMACRXCLIENTCLKIN
- CLIENTEMACTXCLIENTCLKIN

Not all clock input sources are required in all modes. Required clock input signals must always be driven by the FPGA logic at the correct frequency for the required mode of operation (see [Clock Definitions and Frequencies](#)).

Only the input clocks are collectively responsible for the correct operation of the Ethernet MAC. It is possible to ignore all clock generator output clocks for correct Ethernet MAC operation, providing the input clocks are alternatively derived. However, the clock generator output clocks are provided precisely for this purpose.

## Clock Connections to and from FPGA Logic

The clock logic described here is not optimized for specific user modes. There are cases where two or more clock generator input clocks can be shared from a common source. This can be a common client clock, shared between transmitter and receiver; in other cases, this can be a common transmitter clock shared between both client and physical domains. It is also possible to share common clocks between two or more Ethernet MACs. [Chapter 6, Physical Interface](#), provides clock management information for optimal clock logic efficiency based on the chosen physical interfaces.

This section provides general-purpose clock logic descriptions to aid in the understanding of the Ethernet MAC, and clock sharing is not considered. Two main clocking modes are described:

- [Standard Clocking Scheme](#)
- Advanced Clocking Scheme: [The Clock Enable Advanced Clocking Scheme](#)

### Standard Clocking Scheme

The Virtex®-6 FPGA Embedded Tri-Mode Ethernet MAC standard clocking scheme, as shown in [Figure B-2](#), is identical to the standard clocking scheme used by the Virtex-5 FPGA Embedded Tri-Mode Ethernet MAC.

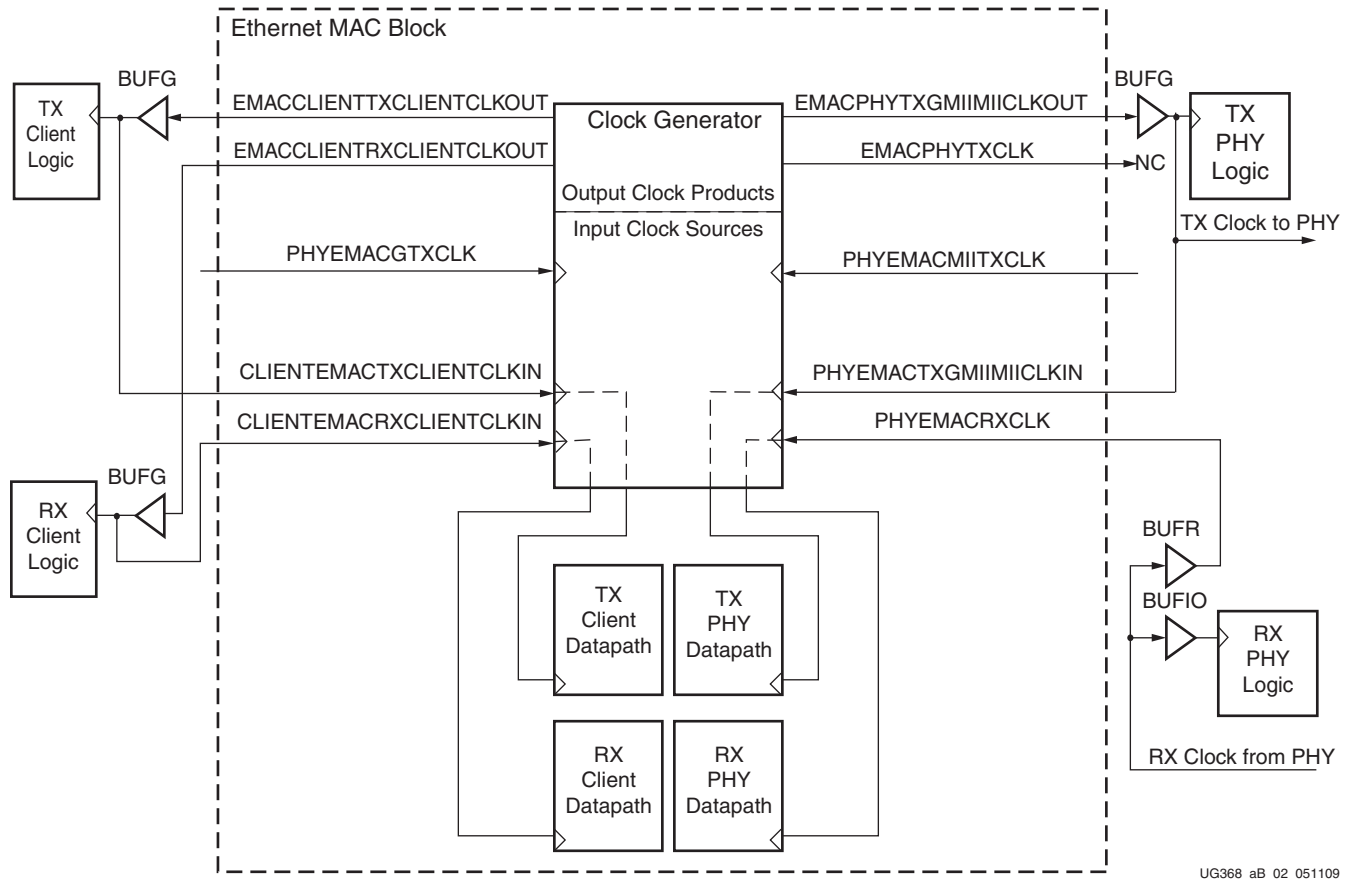


Figure B-2: Overview of Ethernet MAC Clock Connections to and from the FPGA Logic

Figure B-2 shows clock logic instantiated in the FPGA logic using the standard method and shows the general case (non-optimized) clocking scheme.

Figure B-2 and the text that follows describes how the clock signals required by the Ethernet MAC can be derived from the output clocks from the clock generator.

### Transmitter Client Clock

EMACCLIENTTXCLIENTCLKOUT can be placed onto global clock routing and used to clock all of the transmitter client logic. This resultant clock is then fed back into the Ethernet MAC on the CLIENTEMACTXCLIENTCLKIN, where it is used to derive the internal clock used for the transmitter client datapath. This configuration has the effect of eliminating clock skew between the Ethernet MAC and the FPGA logic (caused by the global clock routing), enabling data to be reliably transferred between the two.

### Receiver Client Clock

EMACCLIENTRXCLIENTCLKOUT can be placed onto global clock routing and used to clock all of the receiver client logic. This resultant clock is then fed back into the Ethernet MAC on the CLIENTEMACRXCLIENTCLKIN, where it is used to derive the internal clock used for the receiver client datapath. This configuration has the effect of eliminating clock skew between the Ethernet MAC and the FPGA logic (caused by the global clock routing), enabling data to be reliably transferred between the two.

## Transmitter Physical Clock

EMACPHYTXGMIIICLKOUT can be placed onto global clock routing and used to clock all of the transmitter physical logic. This resultant clock is then fed back into the Ethernet MAC on the PHYEMACTXGMIIICLKIN, where it is used to derive the internal clock used for the transmitter physical datapath. This configuration has the effect of eliminating clock skew between the Ethernet MAC and the FPGA logic (caused by the global clock routing), enabling data to be reliably transferred between the two.

## Receiver Physical Clock

The physical clock for the receiver interface is sourced by the connected PHY. When placed onto global clock routing, this resultant clock is also fed into the Ethernet MAC on the PHYEMACRXCLK port and is used to derive the internal clock used for the receiver physical datapath. This enables data to be reliably transferred from the FPGA logic into the Ethernet MAC.

## The Clock Enable Advanced Clocking Scheme

The Clock Enable advanced clocking scheme reduces the global clock usage in the FPGA logic.

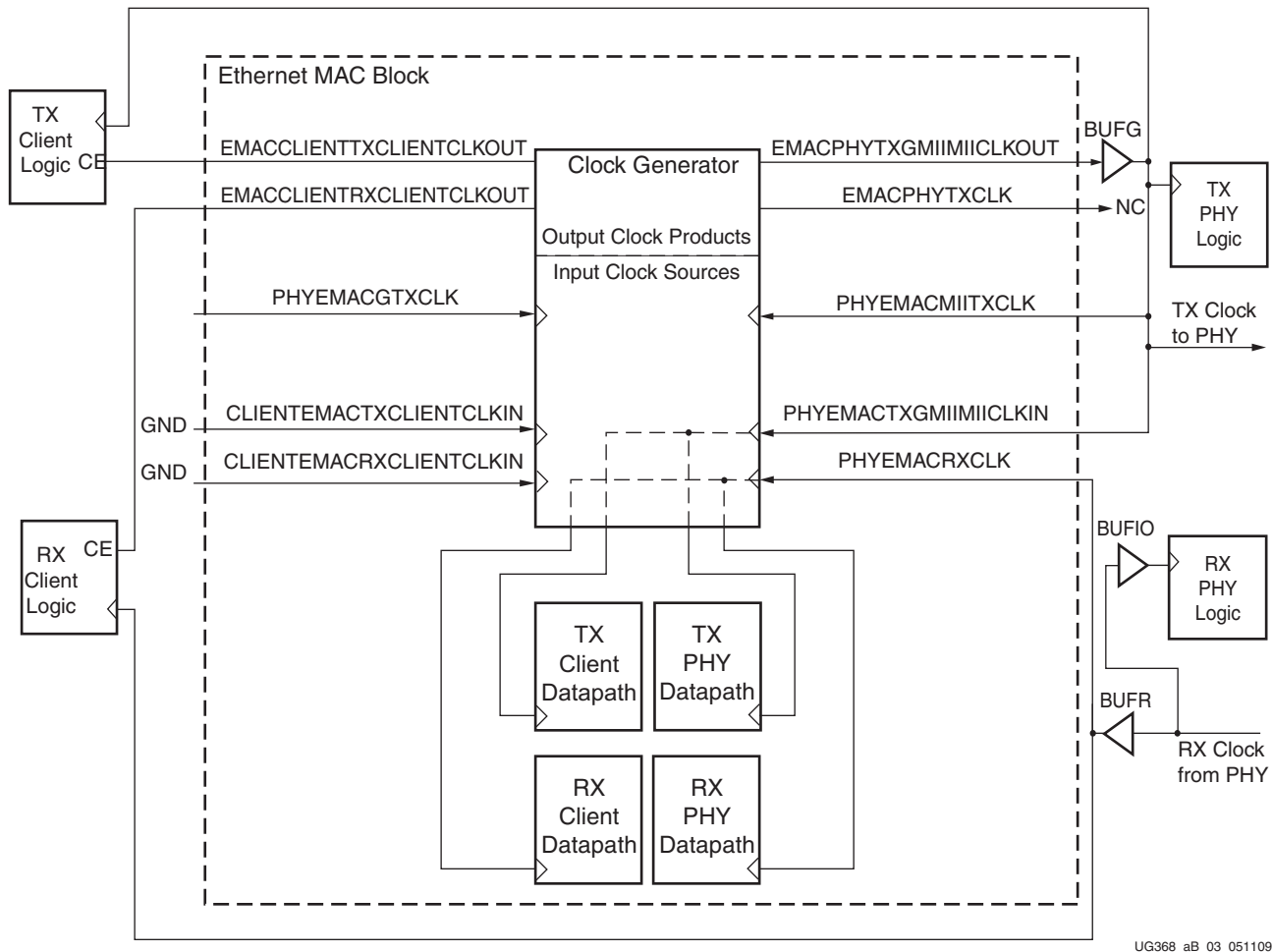
For GMII transmitter operation at 1 Gb/s, both client and physical interfaces have an 8-bit datapath operating at 125 MHz. If 1 Gb/s is the only speed of interest, a single clock domain can be used for both client and physical interfaces.

For MII transmitter operation at 100 Mb/s, the client interface still has an 8-bit datapath that operates at 12.5 MHz. However, the physical interface width dropped to 4 bits and is, therefore, clocked at twice the frequency of the client (25 MHz).

The Clock Enable approach always clocks the client interface with the physical interface clock, while also providing the client with a clock enable. At 1 Gb/s, the physical interface clock is 125 MHz, which is also required by the client; the clock enable is constantly held High. At 100 Mb/s, the physical interface clock is 25 MHz, which is twice that required by the client. The clock enable toggles on alternative 25 MHz clock cycles to enable the client logic at the resultant rate of 12.5 MHz.

This methodology is applied similarly to the receiver path. GMII/MII and RGMII physical interfaces can all use this clocking scheme at all three Ethernet speeds.

Figure B-3 shows clock logic instantiated in the FPGA logic when using the clock enables. Refer to [Clock Definitions and Frequencies, page 199](#) for relationships between clocks and clock enables). Figure B-3 and the text that follows describe how the clock signals required to use the Ethernet MAC can be derived from the output clocks (and clock enables) from the Ethernet MAC clock generator.



UG368\_aB\_03\_051109

**Figure B-3: Overview of Ethernet MAC Clock Connections to and from the FPGA Logic using Clock Enables**

**Note:** There are some complications in the use of the EMACCLIENTTXACK signal when using this Clock Enable scheme. Refer to the specific physical interface section in [Chapter 6](#) for details.

### Transmitter Clock

EMACPHYTXGMIIICLKOUT can be placed onto global clock routing and used to clock all of the transmitter physical logic. This resultant clock is then fed back into the Ethernet MAC on the PHYEMACTXGMIIICLKIN, where it is used to derive the internal clock used for the entire transmitter datapath (both physical and client). This has the effect of eliminating clock skew between the Ethernet MAC and the FPGA logic (caused by the global clock routing), enabling data to be reliably transferred between the two.

EMACCLIENTTXCLIENTCLKOUT is no longer used as a clock but instead as a clock enable. Every flip-flop of the transmitter client logic can be clocked with the clock enable connected to PHYEMACTXGMIIICLKIN, providing it has its clock enable connected as illustrated. This configuration saves global clock resources when compared with the standard clocking approach.

### Receiver Clock

The Physical clock for the receiver interface is sourced by the connected PHY. When placed onto global clock routing, this resultant clock is also fed into the Ethernet MAC on the PHYEMACRXCLK port. This is used to derive the internal clock used for the receiver datapath (both physical and client). This enables data to be reliably transferred from the FPGA logic into the Ethernet MAC.

EMACCLIENTRXCLIENTCLKOUT is no longer used as a clock but instead as a clock enable. Every flip-flop of the receiver client logic can be clocked with the clock connected to PHYEMACRXCLK, providing it has its clock enable connected as illustrated. This configuration saves global clock resources when compared with the standard clocking approach.

## Clock Definitions and Frequencies

The following sections provide definitions for all of the Ethernet MAC input and output clocks, which are often mode dependent. Specific clocking schemes, which are optimized for the chosen physical interface, are provided in [Chapter 6, Physical Interface](#). [Chapter 6](#) provides useful reference material when studying the following sections.

### PHYEMACGTXCLK

#### MII, 1 Gb/s GMII, 1 Gb/s RGMII, and Tri-Speed GMII with Clock Enables Modes

This clock signal is unused and can be connected to logic 0.

#### All Other Modes

For all modes that require it, PHYEMACGTXCLK should be connected to a 125 MHz reference clock source, which is within the IEEE Std 802.3-2005 specification (100 ppm).

**Table B-1: PHYEMACGTXCLK Clock Frequencies**

Clock Signal	Direction	Operating Speed		
		1000 Mb/s	100 Mb/s	10 Mb/s
PHYEMACGTXCLK	Input	125 MHz	125 MHz	125 MHz

### PHYEMACMIITXCLK

#### MII or Tri-Speed GMII Modes

PHYEMACMIITXCLK is used as the transmitter reference clock source when using the MII, where this clock is provided by the attached external PHY device.

**Table B-2: PHYEMACMIITXCLK Clock Frequencies**

Clock Signal	Direction	Operating Speed		
		1000 Mb/s	100 Mb/s	10 Mb/s
PHYEMACMIITXCLK	Input	n/a	25 MHz	2.5 MHz

## 1000BASE-X PCS/PMA (16-Bit Data Client) Mode

When operating in 1000BASE-X PCS/PMA (see [16-Bit Data Client, page 162](#)), this clock input should be connected to a clock source that is half the frequency of the clock input into the CLIENTEMACTXCLIENTCLKIN port.

## All Other Modes

This clock signal is unused and can be connected to logic 0.

## PHYEMACRXCLK

### GMII, MII, and RGMII Modes

The clock signal input into PHYEMACRXCLK is sourced by the external PHY device and should be used by the FPGA logic to clock the receiver physical interface logic. Internally in the Ethernet MAC, this clock is used to derive all physical and client receiver clocks.

Table B-3: PHYEMACRXCLK Clock Frequencies

Clock Signal	Direction	Operating Speed		
		1000 Mb/s	100 Mb/s	10 Mb/s
PHYEMACRXCLK	Input	125 MHz	25 MHz	2.5 MHz

### SGMII and 1000BASE-X PCS/PMA (8-Bit Data Client) Modes

This clock signal is unused and can be connected to logic 0.

## 1000BASE-X PCS/PMA (16-Bit Data Client) Mode

When operating in 1000BASE-X PCS/PMA (see [16-Bit Data Client, page 162](#)), this clock input should be connected to a clock source that is half the frequency of the clock input into the PHYEMACGTCLK port.

## EMACPHYTXGMIIMIICLKOUT, PHYEMACTXGMIIMIICLKIN

### II, GMII, and RGMII Modes

EMACPHYTXGMIIMIICLKOUT is created by the clock generator to provide the frequencies defined in [Table B-4](#).

The clock signal input into the PHYEMACTXGMIIMIICLKIN port should be used to clock the transmitter physical interface logic. This can be derived from EMACPHYTXGMIIMIICLKOUT as illustrated in [Figure B-2](#) and [Figure B-3](#).

Table B-4: EMACPHYTXGMIIMIICLKOUT, PHYEMACTXGMIIMIICLKIN Clock Frequencies

Clock Signal	Direction	Operating Speed		
		1000 Mb/s	100 Mb/s	10 Mb/s
EMACPHYTXGMIIMIICLKOUT	Output	125 MHz	25 MHz	2.5 MHz
PHYEMACTXGMIIMIICLKIN	Input	125 MHz	25 MHz	2.5 MHz



## SGMII and 1000BASE-X PCS/PMA Modes

These clock signals are unused. EMACPHYTXGMIIMIICLKOUT should be left unconnected, and PHYEMACTXGMIIMIICLKIN should be connected to logic 0.

## EMACPHYTXCLK

This clock signal is not used in normal operation and should be left unconnected.

## EMACCLIENTTXCLIENTCLKOUT, CLIENTEMACTXCLIENTCLKIN

### Clock Enables Not in Use

EMACCLIENTTXCLIENTCLKOUT is created by the clock generator to provide the frequencies defined in [Table B-5](#).

The clock signal input into the CLIENTEMACTXCLIENTCLKIN port should be used to clock the transmitter client interface logic. This can be derived from EMACCLIENTTXCLIENTCLKOUT as illustrated in [Figure B-2](#).

**Table B-5: EMACCLIENTTXCLIENTCLKOUT, CLIENTEMACTXCLIENTCLKIN Clock Frequencies**

Clock Signal	Direction	Operating Speed		
		1000 Mb/s	100 Mb/s	10 Mb/s
EMACCLIENTTXCLIENTCLKOUT	Output	125 MHz	12.5 MHz	1.25 MHz
CLIENTEMACTXCLIENTCLKIN	Input	125 MHz	12.5 MHz	1.25 MHz

### Clock Enables in Use

CLIENTEMACTXCLIENTCLKIN is unused and can be connected to logic 0.

EMACCLIENTTXCLIENTCLKOUT is a clock enable rather than a clock. This clock enable is synchronous to the clock input into the PHYEMACTXGMIIMIICLKIN port as illustrated in [Figure B-3](#). All transmitter client logic should be clocked from PHYEMACTXGMIIMIICLKIN and should use EMACCLIENTTXCLIENTCLKOUT as a clock enable as illustrated in [Figure B-3](#).

[Table B-6](#) shows the relationship between the PHYEMACTXGMIIMIICLKIN clock and the EMACCLIENTTXCLIENTCLKOUT clock enable.

**Table B-6: EMACCLIENTTXCLIENTCLKOUT, CLIENTEMACTXCLIENTCLKIN Clock Frequencies**

Clock Signal	Direction	Operating Speed		
		1000 Mb/s	100 Mb/s	10 Mb/s
PHYEMACTXGMIIMIICLKIN	Input	125 MHz	25 MHz	2.5 MHz
EMACCLIENTTXCLIENTCLKOUT	Output	Held at logic 1	Alternates between logic 0 and 1 to produce half clock rate	

## EMACCLIENTRXCLIENTCLKOUT, CLIENTEMACRXCLIENTCLKIN

### Clock Enables Not in Use

EMACCLIENTRXCLIENTCLKOUT is created by the clock generator to provide the frequencies defined in [Table B-7](#).

The clock signal input into the CLIENTEMACRXCLIENTCLKIN port should be used to clock the receiver client interface logic. This can be derived from EMACCLIENTRXCLIENTCLKOUT as illustrated in [Figure B-2](#).

**Table B-7: EMACCLIENTRXCLIENTCLKOUT, CLIENTEMACRXCLIENTCLKIN Clock Frequencies**

Clock Signal	Direction	Operating Speed		
		1000 Mb/s	100 Mb/s	10 Mb/s
EMACCLIENTRXCLIENTCLKOUT	Output	125 MHz	12.5 MHz	1.25 MHz
CLIENTEMACRXCLIENTCLKIN	Input	125 MHz	12.5 MHz	1.25 MHz

### Clock Enables in Use

CLIENTEMACRXCLIENTCLKIN is unused and can be connected to logic 0.

EMACCLIENTRXCLIENTCLKOUT is a clock enable rather than a clock. This clock enable is synchronous to the clock input into the port PHYEMACRXCLK as illustrated in [Figure B-3](#). All receiver client logic should be clocked from the clock input PHYEMACRXCLK and should use EMACCLIENTRXCLIENTCLKOUT as a clock enable as illustrated in [Figure B-3](#).

[Table B-8](#) shows the relationship between the PHYEMACRXCLK clock and the EMACCLIENTRXCLIENTCLKOUT clock enable.

**Table B-8: EMACCLIENTRXCLIENTCLKOUT, PHYEMACMIITXCLK Clock Frequencies**

Clock Signal	Direction	Operating Speed		
		1000 Mb/s	100 Mb/s	10 Mb/s
PHYEMACRXCLK	Input	125 MHz	25 MHz	2.5 MHz
EMACCLIENTRXCLIENTCLKOUT	Output	Held at logic 1	Alternates between logic 0 and 1 to produce half clock rate	

# *Virtex-5 to Virtex-6 FPGA Enhancements*

---

The Virtex®-6 FPGA Embedded Tri-Mode Ethernet MAC is derived from the Virtex-5 FPGA Embedded Tri-Mode Ethernet MAC with some key enhancements and a few minor modifications, as documented in this appendix.

The Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC exists as a single instance within the block. It no longer uses the dual EMAC (referred to as EMAC0/EMAC1) architecture, as was the case for Virtex-5 FPGA Embedded Tri-Mode Ethernet MAC. This modification simplifies configuration and reduces potential resource waste.

## New Features

### Dynamic Switching Between SGMII and 1000BASE-X Standards

Dynamic switching between SGMII and 1000BASE-X physical interface standards is now supported via the host interface (see the Ethernet MAC Mode Configuration Register, [Table 4-6, page 86](#), for register details). This is useful when using off-the-shelf PHY devices with the ability to perform both BASE-X and BASE-T standards, or when a single Virtex-6 FPGA bitstream must be used with either a BASE-X or BASE-T PHY device (see [Switching Between SGMII and 1000BASE-X Standards, page 185](#) for more details).

### Programmable Auto-Negotiation Link Timer

In addition to the programmable auto-negotiation link timer default value set by the EMAC\_LINKTIMERVAL[8:0] attribute, link timer values can be dynamically written via the host interface (see the Ethernet MAC Mode Configuration Register, [Table 4-6, page 86](#), for register details). This function is necessary in support of dynamic switching between SGMII and 1000BASE-X physical interface standards, but may also be useful in other circumstances.

### Up to 2.5 Gb/s Overclocked Operation

When the Ethernet MAC is configured for 1000BASE-X PCS/PMA operation, with 16-bit client interface data width, clock frequencies can be increased in support of a maximum data rate of 2.5 Gb/s. While this can be useful for backplane applications, it should not be used for Ethernet-compliant designs since IEEE Std 802.3-2005 specifies a 1 Gb/s speed. (See [16-Bit Data Client, page 162](#), for more details.)

## Control Frame Length Check Disable

By default, the Ethernet MAC accepts control frames whose length is exactly the minimum legal frame length. Control frames of differing lengths are discarded. Control frame length checking can be disabled, thereby accepting control frames larger than the legal minimum size, via the host interface or the EMAC\_CTRLLENCHECK\_DISABLE attribute. See [Control Frame Length Checks](#), page 62, for more details.

## Ignore MDIO Broadcast Address

The Ethernet MAC's internal PCS/PMA Management Registers respond to the MDIO broadcast address (PHYAD 0) as specified in IEEE Std 802.3-2005. This behavior can be disabled via the EMAC\_MDIO\_IGNORE\_PHYADZERO, as described in [Table 2-18](#), page 41.

## Address Match Statistics Bit

When the Ethernet MAC is configured in Address Filter mode, the Receiver Statistics Vector bit 27, ADDRESS\_MATCH, indicates whether the proceeding frame successfully passed through the address filter. See [Table 3-4](#), page 79.

## Address Filter Changes

To prevent frame corruption, host writes to the address filter registers during the interframe gap take effect immediately, whereas writes during frame reception affect only subsequent frames. Additionally, the address filter no longer accepts frames with a Destination Address of all 0s by default. To accept these frames, an address of all 0s needs to be written to the address filter.

## Hardened Shim

The soft shim logic between the Ethernet MAC and serial transceiver has been immersed into the embedded Ethernet MAC block, and therefore no longer requires FPGA resources to implement.

## Port Map Changes

The following ports were removed:

- PHYEMAC#RXCHECKINGCRC: In Virtex-5 devices, this signal is reserved and not used.
- PHYEMAC#RXBUFERR: In Virtex-5 devices, this signal is reserved and not used.
- PHYEMAC#RXCOMMADET: In Virtex-5 devices, this signal is reserved and not used.
- PHYEMAC#RXLOSSOFSYNC[1:0]: In Virtex-5 devices, this signal is reserved and not used.
- HOSTEMAC1SEL: This signal is unnecessary in the Virtex-6 Ethernet MAC architecture.

The use model of the following port was changed:

- PHYEMACCOL: This signal is tied to GND when the Ethernet MAC is configured for 1000BASE-X or SGMII interfaces. In Virtex-5 devices, it was connected to the TXRUNDISP port of the serial transceiver.

## Attribute Changes

The following attributes were added in support of new features:

- EMAC\_CTRLLENCHECK\_DISABLE: Disables the length check for control frames, allowing frames larger than 64 total bytes.
- EMAC\_MDIO\_IGNORE\_PHYADZERO: Causes the Ethernet MAC's internal PCS/PMA Management Registers to not respond to the MDIO broadcast address (PHYAD 0).

The following attribute was removed:

- EMAC#\_CONFIGVEC79

Additional information can be found in [Chapter 2, Ethernet MAC Overview](#).

## Use Model Changes

The following use models have changed:

- MII and GMII physical interface operation is supported at 2.5V only. Care must be taken to select a PHY that is electrically compatible with the Virtex-6 FPGA LVCMOS25 voltage standard. See the *Virtex-6 FPGA Data Sheet: DC and Switching Characteristics* for more information.
- The Byte PHY advanced clocking scheme is no longer a supported use model.



# *Differences between the Embedded Tri-Mode Ethernet MAC and the Soft IP Cores*

---

This appendix describes the differences between the Embedded Tri-Mode Ethernet MAC block and the soft IP core solutions provided by the CORE Generator™ tool. The functionality provided by the Embedded Tri-Mode Ethernet MAC can also be provided by linking together the Tri-Mode Ethernet MAC soft IP core and the Ethernet 1000BASE-X PCS/PMA or SGMII core. More details are available at:

[http://www.xilinx.com/products/design\\_resources/conn\\_central/protocols/gigabit\\_ethernet.htm](http://www.xilinx.com/products/design_resources/conn_central/protocols/gigabit_ethernet.htm)

There are, however, some differences in the operation of the Embedded Tri-Mode Ethernet MAC and the soft IP cores. These differences are detailed below.

## **Features Exclusive to the Embedded Tri-Mode Ethernet MAC**

These features are exclusive to the Embedded Tri-Mode Ethernet MAC:

- Includes integrated SGMII and 1000BASE-X PCS/PMA functionality.
- Supports up to 2.5 Gb/s overclocked operation in 1000BASE-X mode, when configured with a 16-bit client interface.
- Includes an RGMII/SGMII status register. See [Table 4-7, page 87](#).
- Can be configured using the DCR bus for embedded processor support.

## **Features Exclusive to Soft IP Cores**

These features are exclusive to soft IP cores:

- The soft Tri-Mode Ethernet MAC core:
  - Supports 1 GB half duplex mode for parallel physical interfaces.
  - Has parallel statistics outputs.

- The soft PCS/PMA core:
  - Supports the ten bit interface (TBI).
  - Outputs a status vector with bits that indicate:
    - The status of the link.
    - The status of the link synchronization state machine.
    - When the core is receiving /C/ ordered sets.
    - When the core is receiving /I/ ordered sets.
    - When the core is receiving invalid data.
- Soft PCS/PMA and Tri-Mode Ethernet MAC cores:
  - Can be connected together to provide a single Ethernet interface, similar to the solution provided by the Embedded Tri-Mode Ethernet MAC.
  - Can be configured by a vector when the management interface is not required. The vector signals equate to attributes in the Embedded Tri-Mode Ethernet MAC.
  - Can be targeted to a wide variety of Xilinx<sup>®</sup> devices. The Embedded Tri-Mode Ethernet MAC is available in the Virtex<sup>®</sup>-6 FPGA, the Virtex-5 FPGA (LXT/SXT/FXT/TXT families), and the Virtex-4 FPGA (FX family).