Ethernet-110 Core

Technical Manual

LSI LOGIC ®

R14004.A

Document Number DB14-000014-01, First Edition (November 1997)
This document describes LSI Logic Corporation's Ethernet-110 Core and will remain the official reference source for all revisions/releases of this product until rescinded by an update.

To receive product literature, call us at 1.800.574.4286 (U.S. and Canada); +32.11.300.531 (Europe); 408.433.7700 (outside U.S., Canada, and Europe) and ask for Department JDS; or visit us at http://www.lsilogic.com.

LSI Logic Corporation reserves the right to make changes to any products herein at any time without notice. LSI Logic does not assume any responsibility or liability arising out of the application or use of any product described herein, except as expressly agreed to in writing by LSI Logic; nor does the purchase or use of a product from LSI Logic convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual property rights of LSI Logic or third parties.

Contents

Preface

Chapter 1 **Introduction**

Chapter 2 **Signal Descriptions**

Figures

Tables

Preface

This manual is the primary reference for the Ethernet-110 (E-110) core, which includes the media access controller (MAC) core, the optional MAC control module core, and the optional media independent interface management (MIIM) core. It contains a complete functional description for the E-110 core and includes complete physical and electrical specifications. The E-110 core is capable of operating at 10 or 100 Mbit/s for Ethernet and Fast Ethernet applications.

Audience

This manual assumes that you have some familiarity with Ethernet protocols and related support devices. It also assumes some familiarity with IEEE standards 802.3/802.3u and all applicable clauses describing 10- and 100-Mbit/s Ethernet operation. The people who benefit from this technical manual are:

- Engineers and managers who are evaluating the E-110 core for use in an ASIC application
- Engineers who are designing the E-110 core into an ASIC

Organization

This document has the following chapters:

- Chapter 1, **Introduction**, provides an introduction to LSI Logic Corporation’s E-110 core.
- Chapter 2, **Signal Descriptions**, provides a detailed description of each of the signals associated with the core.
- Chapter 3, **Core Description**, describes the operation of the E-110 core.

- Chapter 4, **Functional Timing**, provides timing diagrams for the core.
- Chapter 5, **Specifications**, describes the AC timing and core inputs and outputs.
- Appendix A, **Glossary**, provides a list of terms used in this manual.

Related Publications

IEEE Standard 802.3, *10 Mbit/s Ethernet Specification*.

IEEE Standard 802.3u, *100BASE-T Fast Ethernet Specification*.

IEEE Standard 802.3x, *802.3 Full Duplex Operation Specification*.

Ethernet-10 (E-10) Core Technical Manual, LSI Logic Corporation, order number R14006.

Conventions Used in This Manual

The first time a word or phrase is defined in this manual, it is *italicized*. See the glossary at the end of this manual for definitions of italicized words.

The word *assert* means to drive a signal true or active. The word *deassert* means to drive a signal false or inactive.

The following signal naming conventions are used throughout this manual:

- A level-significant signal that is true or valid when the signal is LOW always has a suffix of _L attached to its name.
- An edge-significant signal that initiates actions on a HIGH-to-LOW transition always has a suffix of _L attached to its name.

Hexadecimal numbers are indicated by the prefix “0x” before the number—for example, 0x32CF. Binary numbers are indicated by the prefix “0b,” for example, 0b0011.0010.1100.1111.

Chapter 1
Introduction

This chapter describes the overall functions of the E-110 core. It contains the following sections:

1.1 CoreWare® Program

An LSI Logic core is a fully defined and reusable block of logic. It supports industry-standard functions, and has predefined timing and layout. Every core possesses significant intellectual property value that comes from the construction, optimization, and productization of a particular function.

Through the CoreWare program, you can create systems on a chip uniquely suited to your applications. The CoreWare library contains a wide range of complex cores targeting the communications, consumer, and computer markets. The library consist of MIPS microprocessor cores, high-speed interconnect cores, MPEG-2 decoder cores, PCI cores, and many more.

The CoreWare library also includes megafunctions and building blocks, which provide useful functions for developing a system on a chip.

Each core has an associated set of deliverables, including:

- RTL simulation models for the Verilog HDL environment
- Structural simulation models for Verilog HDL and VHDL environments
- A system verification environment (SVE)[1] for RTL-based Verilog simulation
- A test vector suite

Because your design requirements are unique, LSI Logic is flexible in working with you to develop your system-on-a-chip CoreWare design. Three different working relationships are available:

- You provide LSI Logic with a detailed specification and LSI Logic does all of the design.
- You design some functions while LSI Logic provides you with the cores and megafunctions, and LSI Logic completes the integration.
- LSI Logic provides you with the cores, megafunctions, and development tools for you to do the entire design and integration.

Whatever the work relationship, LSI Logic's advanced CoreWare methodology and ASIC process technologies consistently produce Right-First-Time™ silicon.

1.2 Fast Ethernet (100 Mbit/s)

The E-110 core is a 100BASE-T *Fast Ethernet* solution. 100BASE-T is one of a number of technologies that provide greater bandwidth and improved client/server response times. The 100BASE-T standard is designed to provide a smooth evolution from 10BASE-T Ethernet—the dominant 10-Mbit/s network used today—to high-speed 100-Mbit/s performance.

1.2.1 Fast Ethernet Overview

The Fast Ethernet standard for 100BASE-T has been established by the *IEEE 802.3* committee, the same committee that developed the original

[1] The SVE provides a software environment for verifying core functionality.

Ethernet standard. The 100BASE-T standard is found in the *IEEE 803.3u* specification, which serves as a supplement to the 802.3 standard and extends the operating speed to 100 Mbit/s.

100BASE-T Fast Ethernet is an extension of 10BASE-T technology. This technology uses the existing 802.3 media access control (MAC) layer (layer 2), connected through a media-independent interface (MII) to a physical layer device (PHY).

The MII specification is analogous to the 10-Mbit/s Ethernet *attachment unit interface (AUI)*—it provides a single interface that can support external transceivers for any of the 100BASE-T media specifications. In the Fast Ethernet MAC, the bit time (the time to transmit one bit of data) has been reduced by a factor of 10, thereby increasing *packet* speed to ten times that of 10BASE-T, while packet format and length, error control, and management information remain identical to 10BASE-T.

The 100BASE-T standard supports three PHY layers:

- 100BASE-TX (see IEEE 802.3u, clauses 24 and 25)
- 100BASE-T4 (see IEEE 802.3u, clause 23)
- 100BASE-FX (see IEEE 802.3u, clauses 24 and 26)

The 100BASE-X standard outlined in the IEEE 802.3u specification embodies the 100BASE-TX and 100BASE-FX standards. The term 100BASE-X is used when referring to the physical coding sublayer (PCS) and physical medium attachment (PMA), which are common to both 100BASE-TX and 100BASE-FX. See Figure 1.2 on page 1-7.

The physical medium dependent (PMD) issues are different for 100BASE-TX and 100BASE-FX, and are described in separate sections of the IEEE 802.3u specification. The 100BASE-T4 standard is unique from 100BASE-TX and 100BASE-FX and is described in its own section of the IEEE 802.3u specification.

The 100BASE-T standard also defines a universal *hub* interface and a management interface.

The cable and fiber types currently used in 100BASE-T networks are:

- 100BASE-TX— a two-pair system for data grade (EIA 568 Category 5) unshielded *twisted-pair* (UTP) and 150-Ω shielded twisted-pair (STP) cabling

- 100BASE-T4—a four-pair system for both voice and data grade (Category 3, 4, or 5) UTP cabling
- 100BASE-FX— two multimode fibers

Additionally, the 100BASE-TX, 100BASE-T4, and 100BASE-FX systems can be mixed and interconnected through a hub.

1.2.2 Fast Ethernet Features

The key advantages of Fast Ethernet:

- Proven technology with speed increased by a factor of 10
- Simple migration from 10-Mbit/s to 100-Mbit/s networks
- Extensive multivendor support

1.2.2.1 10BASE-T to 100BASE-T Comparison

Fast Ethernet, in the form of 100BASE-T technology, is really very similar to 10BASE-T, but is ten times faster. Unlike other high-speed technologies, Ethernet has been installed for over 20 years in business, government, and educational networks. Table 1.1 is a comparison between 10BASE-T and 100BASE-T.

Table 1.1 10BASE-T Ethernet vs. 100BASE-T Fast Ethernet

Function	Ethernet	Fast Ethernet
Speed	10 Mbit/s	100 Mbit/s
IEEE Standard	802.3	802.3u
Media Access Protocol	CSMA/CD	CSMA/CD
Topology	Bus or Star	Star
Cable Support	Coax, UTP, Fiber	UTP, Fiber
UTP[1] Cable Support	Category 3, 4, or 5	Category 3, 4, or 5
UTP Link Distance (max)	100 Meters	100 Meters
Collision Domain Diameter (maximum with all UTP)	500 Meters	205 Meters
Maximum Network Diameter (using switches/routers)	Unlimited	Unlimited
Media Independent Interface?	Yes (AUI)[2]	Yes (MII)[3]
Full Duplex Capable?	Yes	Yes

1. UTP = unshielded twisted pair
2. AUI = auxiliary unit interface
3. MII = media independent interface

1.2.2.2 CSMA/CD Transmission Protocol

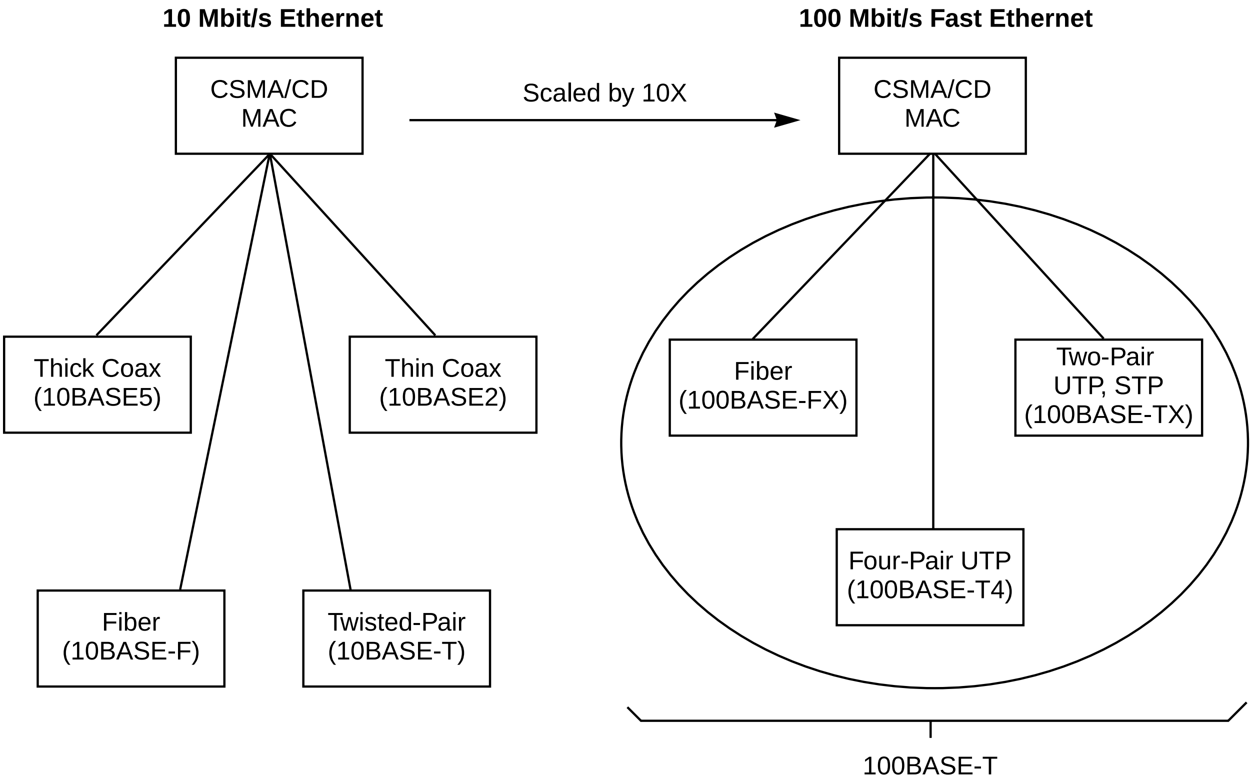
Ethernet's fundamental transmission protocol is *Carrier Sense Multiple Access with Collision Detection (CSMA/CD)*. Fast Ethernet maintains this protocol, which allows it to move data between 10BASE-T and 100BASE-T stations on a *local area network (LAN)* without requiring protocol translation. A simple, low-cost *switching hub* or *bridge* is all that is required for connection. Because it is the same technology (but faster), 100BASE-T can be successfully integrated into 10BASE-T networks to form a reliable migration path to 100BASE-T networks.

1.2.2.3 Media Specifications

A major advantage of Fast Ethernet over other high-performance technologies is that it can be deployed as a switching or shared technology. 100BASE-T networks are based on a combination of switching and shared hubs in order to maintain existing 10BASE-T infrastructures. 100BASE-T networks are capable of full-duplex operation.

100BASE-T combines the scaled MAC with a variety of media specifications to provide users with the maximum possible wiring flexibility, as shown in Figure 1.1. The media specifications (100BASE-TX, 100BASE-T4, and 100BASE-FX) are collectively referred to as 100BASE-T and enable users to retain their existing cabling infrastructure while migrating to Fast Ethernet.

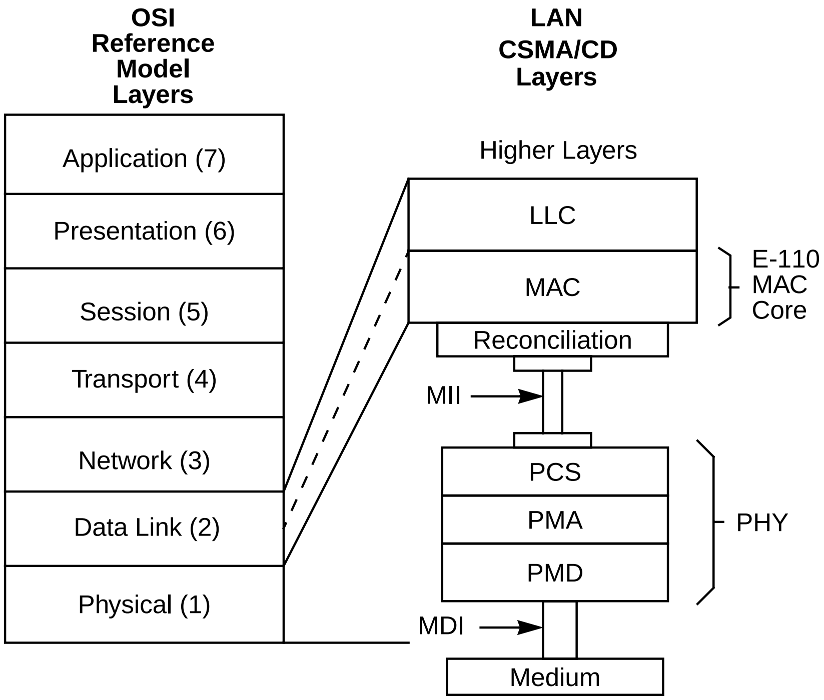
Figure 1.1 100BASE-T Media Specifications



1.2.3 E-110 MAC Core and the OSI Model

Figure 1.2 shows how the E-110 MAC fits into the ISO Open Systems Interconnection (OSI) Reference model. The OSI model defines a data communications protocol consisting of seven distinct layers.

Figure 1.2 E-110 MAC Relationship to the OSI Model



LLC = logical link control
MAC = media access control
MDI = media-dependent interface
MII = media-independent interface

PCS = physical coding sublayer
PHY = physical layer device
PMA = physical medium attachment
PMD = physical medium dependent

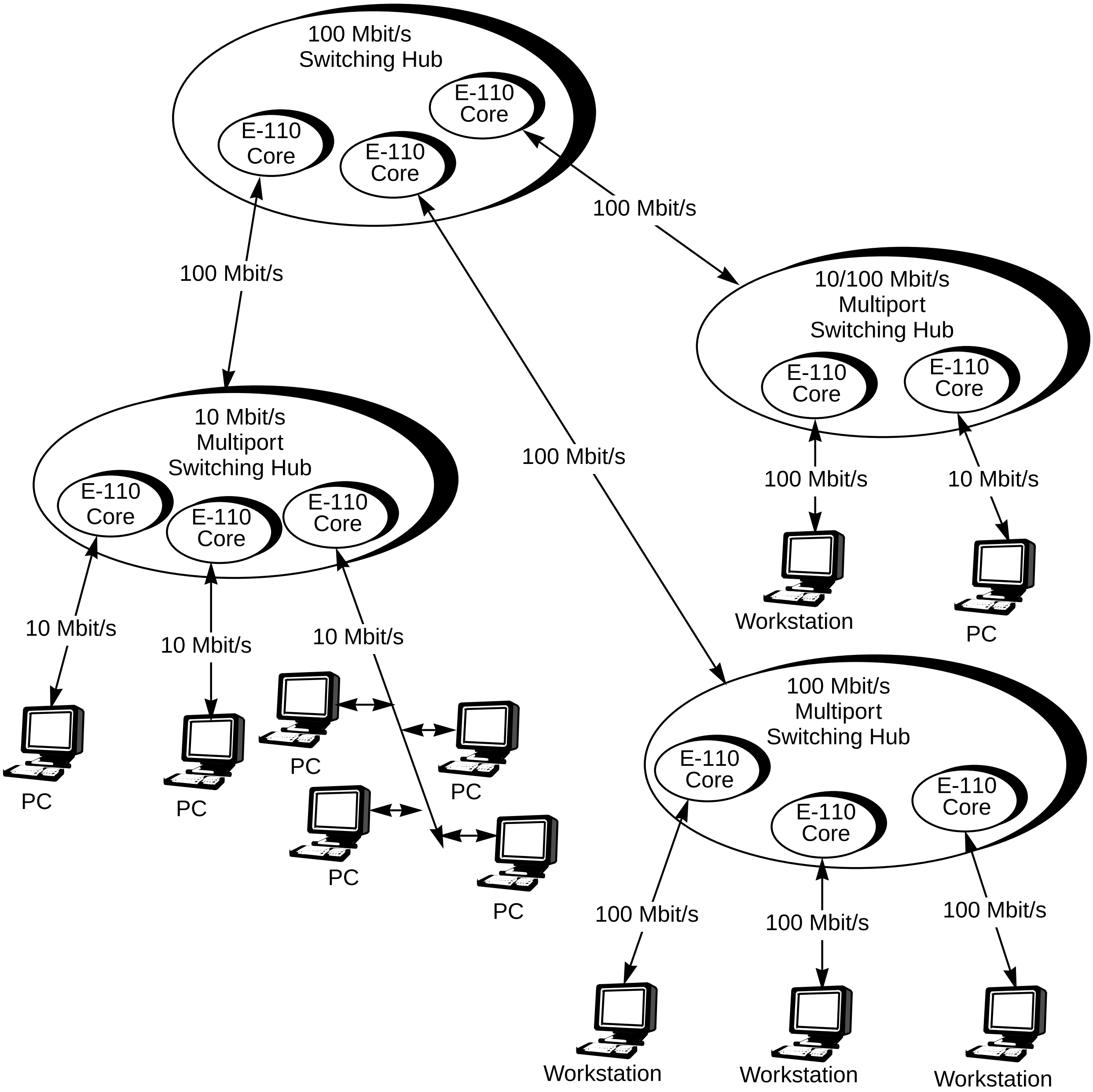
Overviews of the operation of the MAC core, MAC control module core, and the MIIM core are found later in this chapter.

1.2.4 Fast Ethernet Hubs

A hub is a common wiring point for star-topology networks, and can perform various functions, such as switching, statistics gathering, and signal retiming.

The E-110 core provides a high-integration solution to hub design because several cores may be placed on a single ASIC. Figure 1.3 illustrates how hubs might be configured using the E-110 core.

Figure 1.3 Hub Examples



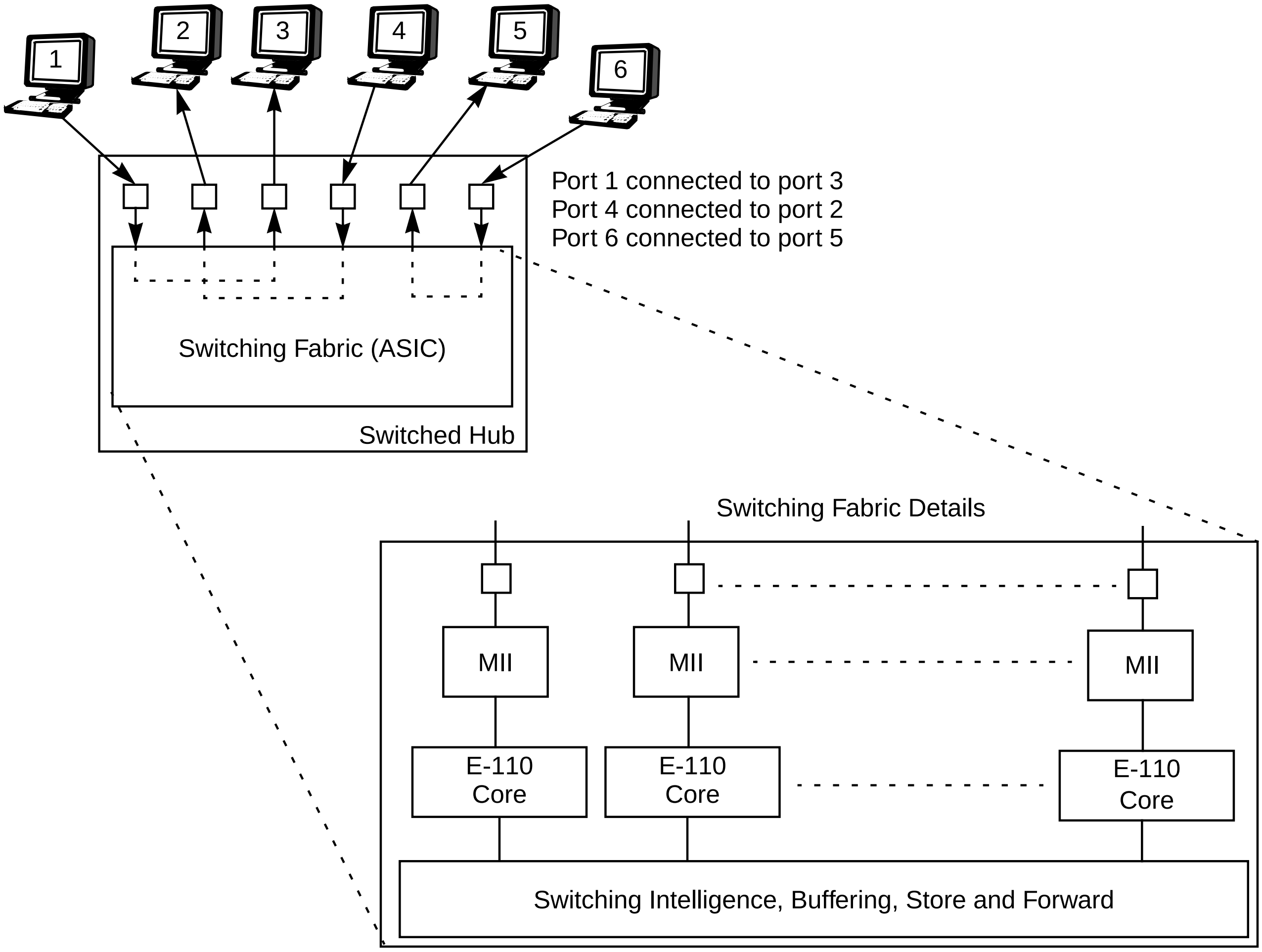
A switching hub facilitates packet switching. Each port of a packet switching hub provides a connection to an Ethernet media system that operates as a separate *Ethernet LAN*. Unlike a repeater hub whose individual ports combine segments together to create a single large LAN, a switching hub makes it possible to divide a set of Ethernet media systems into multiple LANs that are linked together by way of the packet switching electronics in the hub. The round trip timing rules for each LAN stop at the switching hub port, allowing you to link a large number of individual Ethernet LANs together. A given Ethernet LAN can consist of a single cable segment linking some number of computers, or it may consist of a repeater hub linking several such media segments together. Whole Ethernet LANs can themselves be linked together to form extended network systems using packet switching hubs. While an individual Ethernet LAN may typically support anywhere from a few up to several dozen computers, the total system of Ethernet LANs linked with packet switches at a given site may support many hundreds or thousands of machines.

A switching hub can be designed with an intelligent switching fabric and multiple E-110 cores. The switching may be done in any of the following manners:

- Cut-Through—the head of the *frame* leaves the switch before the tail has arrived. You use an address compare buffer to decode the destination address.
- Store and Forward—the entire frame is received and buffered before being forwarded to an output port.
- Adaptive Cut-Through—the hub looks at *collision* statistics. If too many collisions occur, the hub changes from cut-through operation to store and forward operation.

Figure 1.4 shows where the E-110 core fits in a switched hub.

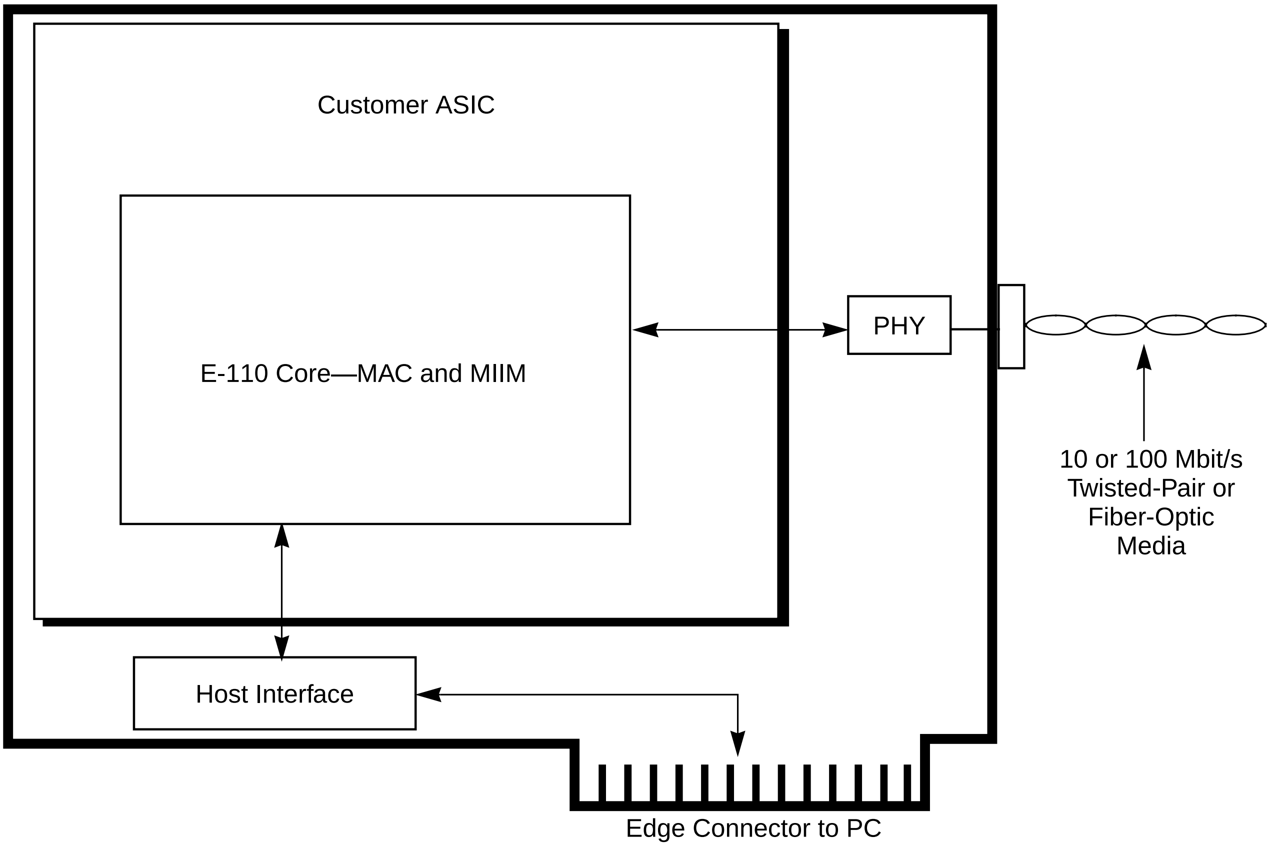
Figure 1.4 Switched Hub Example


1
2
3
4
5
6
Port 1 connected to port 3
Port 4 connected to port 2
Port 6 connected to port 5
Switching Fabric (ASIC)
Switched Hub
Switching Fabric Details
MII
MII
MII
E-110 Core
E-110 Core
E-110 Core
Switching Intelligence, Buffering, Store and Forward

1.2.5 Fast Ethernet Network Interface Cards

Network cards can be designed with the E-110 core that will run at speeds of 10 or 100 Mbit/s. Figure 1.5 shows how the E-110 core fits into a network interface card (NIC). The E-110 core provides the MAC function and interfaces with 10 Mbit/s or 100 Mbit/s PHYs for single- or multi-point server solutions.

Figure 1.5 Network Interface Card Block Diagram

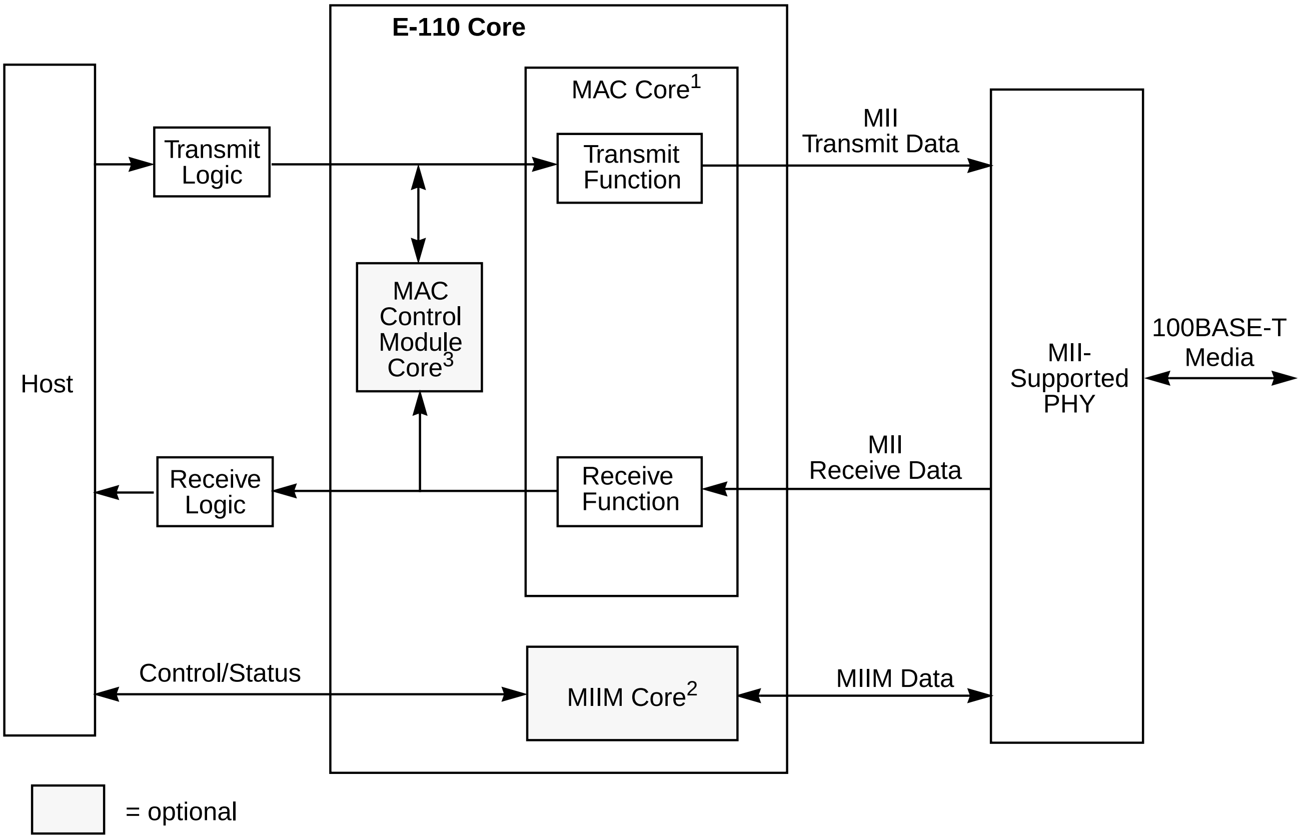


1.3 E-110 Core

As shown in Figure 1.6, the E-110 core contains the following cores:

- MAC
- MAC Control Module (optional)
- MIIM (optional)

Figure 1.6 E-110 ASIC Environment



1. The MAC is capable of 10- or 100-Mbit/s operation.
2. An optional MIIM core reads PHY status registers and writes PHY control registers.
3. An optional MAC control module core provides the flow control functions.

The MAC[1] core operates at 10 or 100 Mbit/s using an external PHY. An optional MAC control module core located external to the E-110 core allows the host to perform the flow control functions given in the IEEE 802.3/802.3u standard. An optional MIIM core allows a host to communicate with the control and status registers within an MII-supported PHY.

[1] The MAC function meets the requirements of the IEEE 802.3u specification.

1.4 E-110 MAC Core

The E-110 MAC core consists of the transmit and receive functions.

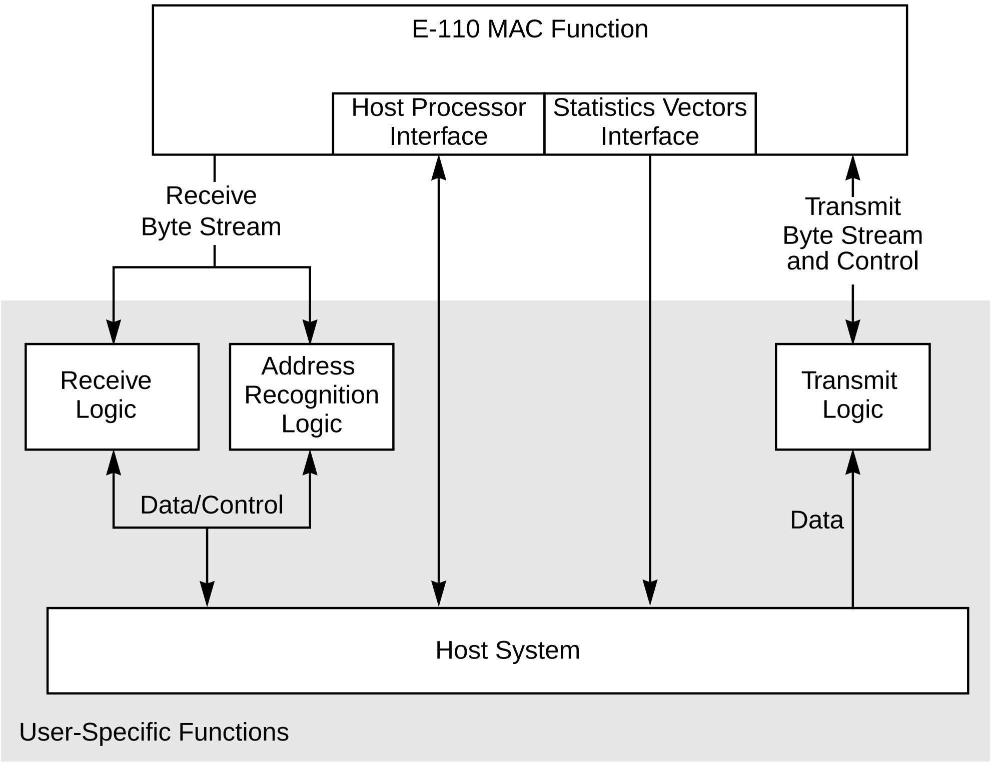
The transmit and receive functions transport data between the MAC function’s host interface and the MII. Data consists of bytes on the host side and nibbles on the MII side. The host provides some data buffering and logic to manage properly the receive and transmit data streams. The MII is a logical data interface only. *The user must provide line drivers and receivers to meet cable interface or attachment unit interface requirements.*

1.4.1 MAC Core Host System Connection

The E-110 MAC core is connected to a host system as shown in Figure 1.7. The following host connections are explained in this section:

- System Data Flow
- Statistics Vectors Interface
- Host Processor Interface

Figure 1.7 E-110 MAC Host System Connection



1.4.1.1 System Data Flow

The E-110 MAC operates at either 10 or 100 Mbit/s between an MII on one side and a host on the other. The receive and transmit data streams are independent from each other on both sides of the MAC.

The receive byte stream from the E-110 MAC operates on the MII receive clock. Another element of the overall system may assemble bytes from the receive byte stream into words for storage in a buffer memory or transmission to a system bus. Alternatively, the receive byte stream might enter directly into a FIFO. The output of the FIFO could then be interfaced into the overall system in a variety of ways, possibly using a common system clock.

The transmit byte stream for the E-110 MAC operates on the MII transmit clock. Another element of the overall system may disassemble words read from a buffer memory or received from a system bus into bytes for the transmit byte stream. Alternatively, the transmit byte stream might exit directly from a FIFO. The input of the FIFO could then be fed from the overall system in a variety of ways, possibly using a common system clock.

Logic associated with the transmit and receive byte streams handles the packet and data flow control between the overall system and the E-110 MAC. Specifically, a device external to the E-110 MAC must monitor the receive byte stream to perform receive address recognition.

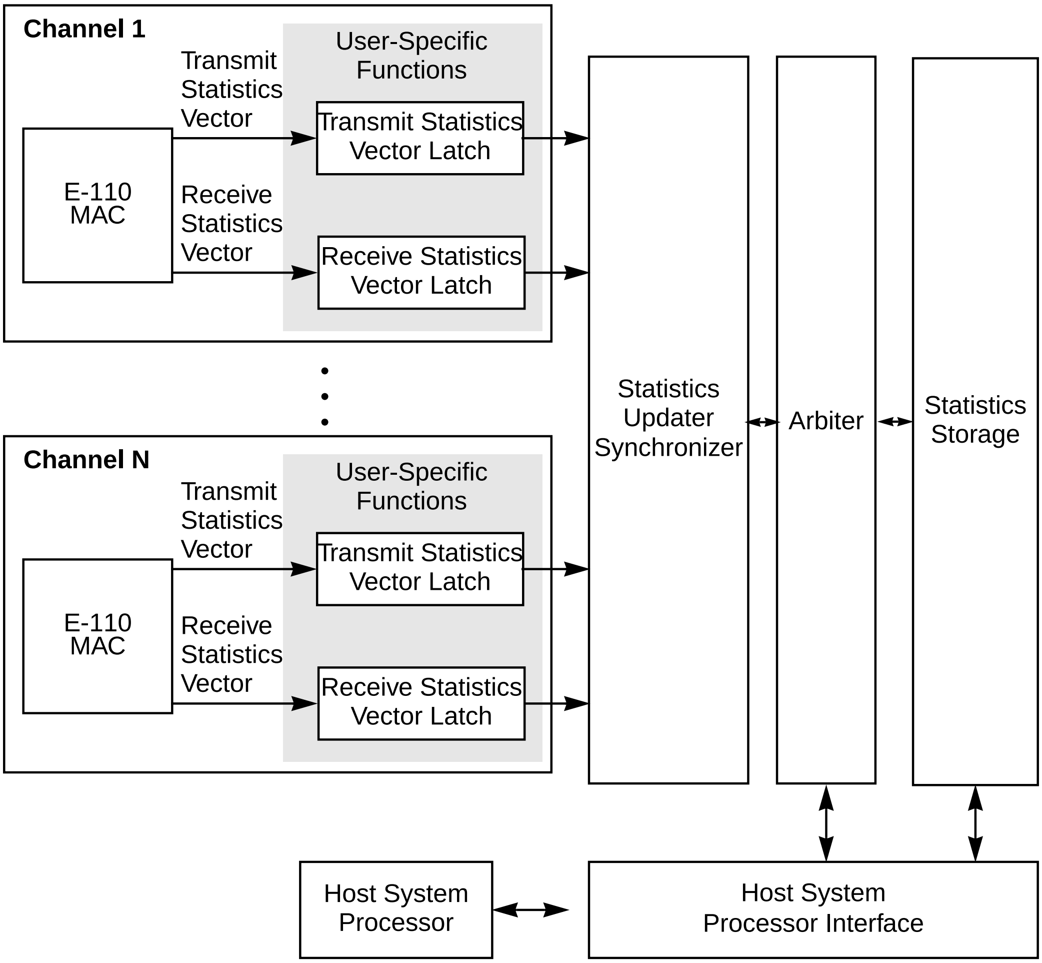
Details of the receive byte stream interface can be found in Section 3.2.1.2, “MAC Receive Function,” on page 3-13. An explanation of the transmit byte stream interface can be found in Section 3.2.1.1, “MAC Transmit Function,” on page 3-5. E-110 MAC behavior can be altered with external control signals. A description of the control signals is given in Chapter 2.

1.4.1.2 Statistics Vectors Interface

The E-110 MAC collects packet-by-packet statistics, but does not include statistics event counters or accumulating byte counters. Statistics for management purposes are collected outside the E-110 MAC for either end station or multiple channel applications.

Figure 1.8 shows how a centralized statistics collector for use with multiple E-110 MACs might be constructed.

Figure 1.8 Centralized Statistics Updater



Channel 1
User-Specific Functions
Transmit Statistics Vector
Transmit Statistics Vector Latch
E-110 MAC
Receive Statistics Vector
Receive Statistics Vector Latch
Channel N
User-Specific Functions
Transmit Statistics Vector
Transmit Statistics Vector Latch
E-110 MAC
Receive Statistics Vector
Receive Statistics Vector Latch
Statistics Updater Synchronizer
Arbiter
Statistics Storage
Host System Processor
Host System Processor Interface

Whenever the E-110 MAC receive and transmit functions complete the processing of a packet, the MAC creates a transmit or a receive statistics vector that may be latched external to the MAC function for use by the statistics updating process. Because the vector is latched, the receive and transmit functions do not hold the statistics vectors indefinitely. It then becomes the responsibility of the host to free up enough statistics storage space before the MAC overwrites the vectors with those for the next processed packet.

When the MAC latches the receive and transmit statistics vectors, the MAC provides synchronized latching signals for each of the transmit and receive vectors. The vectors include Packet Length bits that specify the length of the packet in bytes. In the example block diagram of Figure 1.8, the Statistics Updater Synchronizer feeds the Arbiter, which schedules the updating of statistics in the Statistics Storage. The statistics subsystem usually operates on the host system microprocessor clock, which means that accesses to the statistics storage by the host are automatically synchronized. However, the transmit and receive statistics vector latch pulse must be synchronized to the host system clock because the transmit and receive functions operate on their own clocks.

1.4.1.3 Host Processor Interface

The host system processor has the following connections to the E-110 MAC:

- ♦ MII Data Interface
- ♦ Random Number Generator Interface
- ♦ Control Interface
- ♦ Statistics Interface

MII Data Interface – The E-110 MAC sends and receives data to the MII-supported PHY by means of the MII Transmit Data and Receive Data signal lines, over which the MAC passes transmit and receive nibbles.

Random Number Generator Interface – The E-110 MAC contains a random number generator that is used for transmit *backoff* in the event of a collision. The host may optionally load an 11-bit seed value into the random number generator. By loading different seed values into each E-110 MAC in a multiple-MAC system, you can guarantee that the backoff timers in each MAC are not synchronized to each other, which avoids the problem of two or more MACs colliding repeatedly while attempting to transmit. If you do not load a seed value in the random number generator, there is a chance that two MACs might have the same backoff timer value.

Control Interface – The host control interface allows the host to directly control the operation of the E-110 MAC. The host must provide either a 25- or 33-MHz clock (HCLK) to the core. The core uses the clock select (CLKS) input signal from the host to determine how to divide down HCLK to create the MIIM Data Clock (MDC) signal. The host provides a variety of control signals to the MAC function to control the following:

- ♦ Interpacket gaps (see Section 3.2.1.1, “MAC Transmit Function,” and the signal descriptions for IPGR1[6:0], IPGR2[6:0], and IPGT[6:0] in Chapter 2).

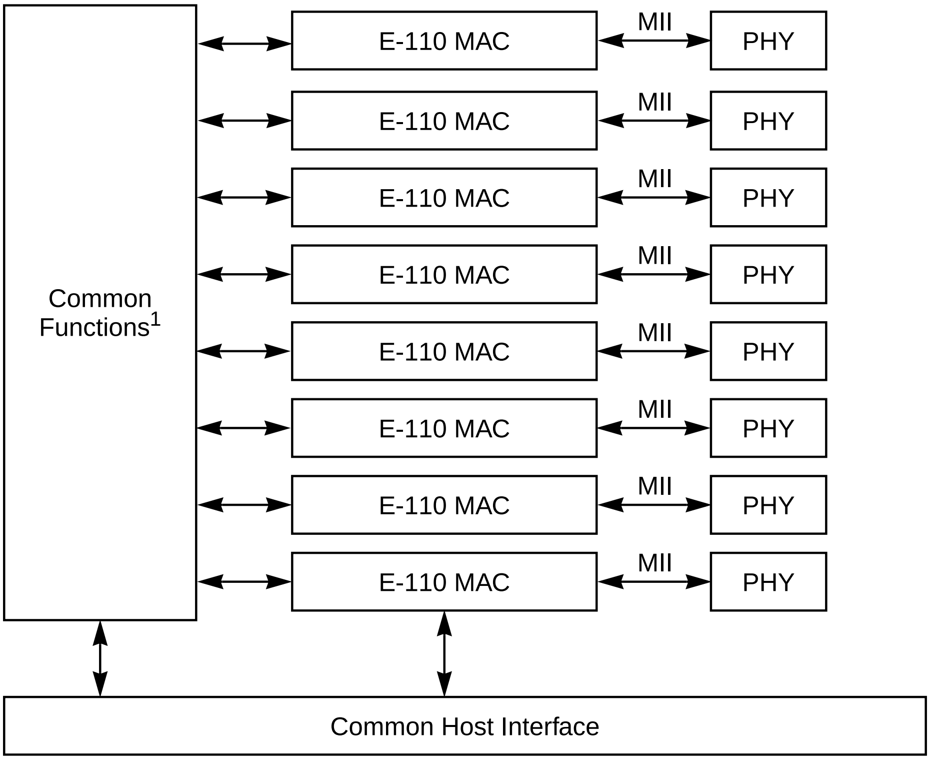
- Padding of packets (see the subsection entitled “IPG Full-Duplex Operation” and the signal descriptions for NOPRE and PADEN in Chapter 2).
- Preamble (see the subsection entitled “IPG Full-Duplex Operation” and the signal description for NOPRE in Chapter 2).
- Full duplex operation (see the subsection entitled ”IPG Full-Duplex Operation," the signal description for FULLD under Section 2.1.4, “MII to PHY Signals,” and the signal description for FULLD in Chapter 2).
- *Frame check sequence (FCS)* checking (see the subsection entitled ”IPG Full-Duplex Operation," Section 3.2.1.2, “MAC Receive Function,” the subsection entitled ”Receive Function to Host Interface," and the signal descriptions for CRCO[9:1], CRCG, CRCEN, and NOPRE in Chapter 2).
- Maximum packet length (see the subsection entitled ”IPG Full-Duplex Operation," Section 3.2.1.2, “MAC Receive Function,” and the signal description for HUGEN in Chapter 2).
- Late collisions (see the subsection entitled ”IPG Full-Duplex Operation," and the signal description for TPAB and RTRYL in Chapter 2).
- 10- or 100-Mbit/s operation (see Table 3.2).

Statistics Interface – The E-110 MAC function provides receive and transmit statistics data that can be latched by a device external to the function and then read by the host. The MAC function provides the information on a parallel interface along with strobe signals.

1.4.2 MAC Multiple Channel Operation

While the E-110 MAC function can be used in a single-channel environment as shown in Figure 1.6, the same design can also be used for multiple channel applications as shown in Figure 1.9.

Figure 1.9 Multiple Channel Application



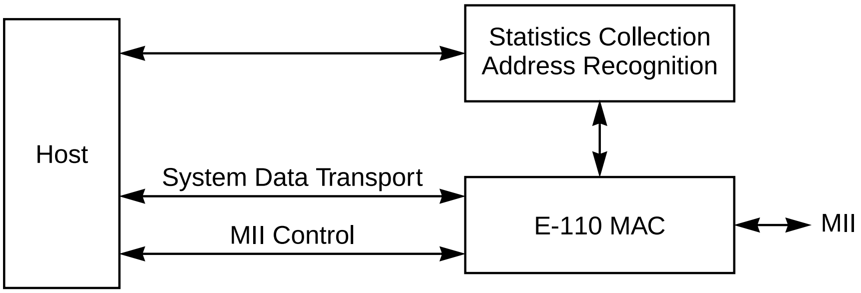
[1] Data transport, address recognition, and statistics collection.

The exact nature of the common sections in Figure 1.9 is highly application dependent. The MII signal lines are only logical implementations without the line driving capability.

1.4.3 MAC External Interfaces

The E-110 MAC operates at either 10 Mbit/s or 100 Mbit/s between a MII on one side and a system data transport on the other. The system designer can design the E-110 MAC into both end station, switch, and hub environments. For system efficiency in multiple MAC situations, you may provide a module external to the MAC function to collect statistics on MAC events and perform receive packet address recognition. Statistics collection and receive packet address recognition are customer defined for use with the E-110 MAC. Figure 1.10 illustrates the connections to the E-110 MAC function.

Figure 1.10 External Connections to the E-110 MAC



For simplicity of application, all per packet interactions (for example, start of frame and end of frame) between the E-110 MAC and the system take place through the system data transport. Using the system data transport eliminates the need for involvement of a processor at 100 Mbit/s because the E-110 MAC takes over the per packet data transfer responsibility.

The E-110 MAC conforms to existing IEEE 802.3 standards for 10-Mbit/s operation and IEEE 802.3u standards for 100-Mbit/s operation. It is expected that a common application of the E-110 MAC will be as a part of a larger system in user-specific ASICs. The E-110 MAC therefore imposes very little system personality regarding word size, byte order, data flow methodology, or management conventions. Alternatively, the E-110 MAC can be combined with a minimum of logic circuitry connected to off-chip interfaces to configure a unique MAC device.

To allow for a variety of network media types and methods of connection to the media, the MII standard has been adopted for the E-110 MAC. The E-110 MAC uses the MII on a logical basis only. Any implementation of the E-110 MAC can be used to connect copper wire or fiber media by means of a PHY. The PHY can reside either within the same ASIC as the E-110 MAC or external to the ASIC, as long as the PHY interface to the E-110 MAC is MII-compatible.

The E-110 MAC function transmits and receives nibbles of data to and from the media over the MII signal lines. The function handles independent receive and transmit byte streams between MII devices and the host. On the host side of the E-110 MAC, the byte streams can be directed into a system memory or bus structure in a variety of ways, such as shared or separate DMA channels or direct access to multiple port memories and FIFOs. The byte streams can also be assembled into system words as desired.

Management statistics collection is not included in the E-110 MAC. To facilitate statistics collection, the MAC function incorporates statistics vector circuitry to simplify the accumulation of event statistics. If statistics collection logic is desired, internal or external to the ASIC, it must be designed and implemented by the end user.

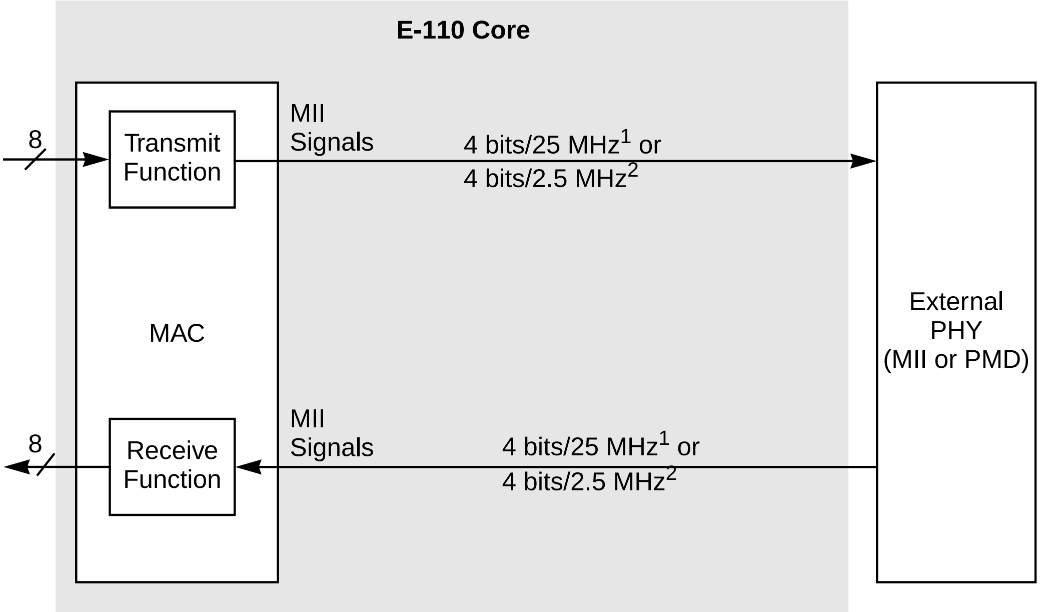
To allow for simple end station and complex switching hub environments, you may add receive packet address recognition logic external to the MAC function. LSI Logic does not supply the address recognition logic. For end station applications, a simple single station address comparison module can be attached to the receive byte stream. A *hashing multicast* capability can also be included in this module to detect sets of destination addresses. For switching hub applications, a central source and destination address filter block is often the preferred solution. The E-110 MAC function transmit and receive byte streams can interact with such a block.

1.4.4 MAC MII

The E-110 MAC operates with the MII signal lines connecting to physical media interface devices. The MII is a logical data interface only. *The designer must provide line drivers and receivers to meet cable interface or AUI requirements.*

Figure 1.11 shows that a processor can control a PHY through an E-110 MAC that incorporates an MII function.

Figure 1.11 MII Function



Notes:

1. 4 bits/25 MHz means that the data path is 4 bits wide (a nibble) and that the clock rate is 25 MHz, which yields 100 Mbit/s
2. 4 bits/2.5 MHz means that the data path is 4 bits wide (a nibble) and that the clock rate is 2.5 MHz, which yields 10 Mbit/s

As new MII PHYs become available, they can be attached to the E-110 MAC, providing increased capability at both 10 Mbit/s and 100 Mbit/s. You can continue to interface to the PHYs by means of the E-110 MAC's built-in MII function.

1.4.5 MAC Backoff Operation

When a data collision occurs, the MAC must perform a backoff operation, which causes the MAC to wait for a time and then try retransmitting. The MAC implements two selectable backoff methods:

- Stop Backoff
- Truncate Backoff

1.4.5.1 Stop Backoff

The Stop Backoff feature is implemented with the TBOFF_SEL input pin. The TBOFF_SEL input, driven by the host, controls whether the E-110 core uses the Binary Exponential Backoff algorithm during collisions. See Chapter 2, “Signal Descriptions,” for a detailed signal description.

The stop backoff implementation does not conform to the IEEE 802.3/802.3u standard, but may be useful for test purposes.

1.4.5.2 Truncate Backoff

The truncate backoff feature is implemented with the BKOFF_LIMIT[1:0] input pins, which are driven by the host. The truncate backoff feature gives the user some flexibility in achieving a performance advantage in some native applications. With the truncate backoff feature, the E-110 core can be configured for either more or less aggressive backoff behavior, depending on the user’s choice. See Chapter 2, “Signal Descriptions,” for a detailed signal description.

1.4.6 MAC Backpressure Feature

The MAC core implements a backpressure feature, which is implemented with the False Carrier Sense (FLS_CRS) input pin. The pin is driven by the host, and is applicable when the E-110 MAC core is used in a half-duplex switched port design.

When using a switch (rather than a repeater) as the hub for an Ethernet LAN, it is possible for traffic to arrive at a switched port faster than the switch is able to transmit the traffic out to the target destination port. Making the channel appear busy to the transmitter is an effective and efficient way of stopping the transmitter from flooding the receiver’s buffers. The use of the FLS_CRS signal is especially clean when only one end station is connected to each switch port—in these cases, FLS_CRS throttles the end station as needed.

See Chapter 2, “Signal Descriptions,” for a detailed signal description of the FLS_CRS signal.

1.5 MAC Control Module Core

The MAC control module is an optional core that interfaces the E-110 core to the host and implements full-duplex flow control.

In full-duplex Ethernet, neither False Carrier Sense nor forced collision algorithms solve congestion control problems. A full-duplex Ethernet station does not implement *collision detection* and ignores Carrier Sense for deferring transmissions. A full-duplex Ethernet implementation therefore requires an explicit flow control mechanism to allow a switch to throttle a congesting end station. Clause 31 in the IEEE 802.3/802.3u standard defines the frame-based flow control scheme implemented in the MAC control module core.

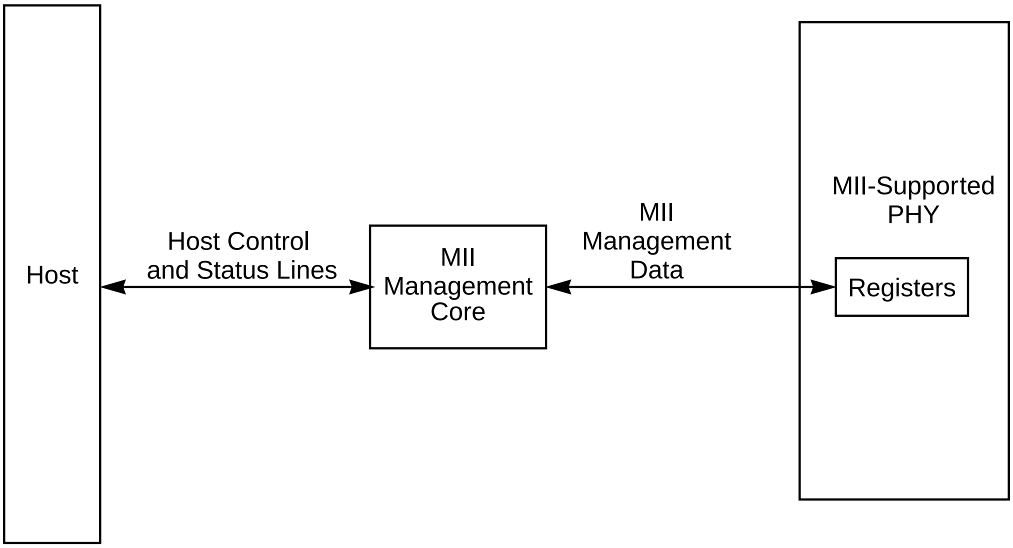
1.6 MIIM Core

The optional MIIM core allows the host to write control data to and read status information from any of 32 registers in any of 32 PHYs using the MIIM interface.

The MIIM interface is a 16-bit parallel interface on the MIIM core's host side and a serial interface on the core's MII side. The MIIM Interface allows the host to send control data and receive status information from PHYs over the MAC MII interface.

The host can select one of 32 PHYs and one of 32 registers within any PHY and then send control data or receive status information. Only one register in one PHY can be addressed at any time. For more details on the communication from the host to the PHYs, refer to the Reconciliation Sublayer and Media Independent Interface Specifications section of the IEEE 802.3u specification, 100BASE-T Fast Ethernet. The host sends control data to the PHY and receives status information from the PHY through the MIIM module, as shown in Figure 1.12.

Figure 1.12 MIIM Core



Chapter 2
Signal Descriptions

This chapter provides detailed descriptions of the signals for the E-110 core. These signal descriptions are useful for designers who will be interfacing the E-110 core with other core logic or external logic. This chapter contains the following sections:

- Section 2.1, “MAC Core Signals”
- Section 2.2, “MAC Control Module Signals”
- Section 2.3, “MIIM Core Signals”

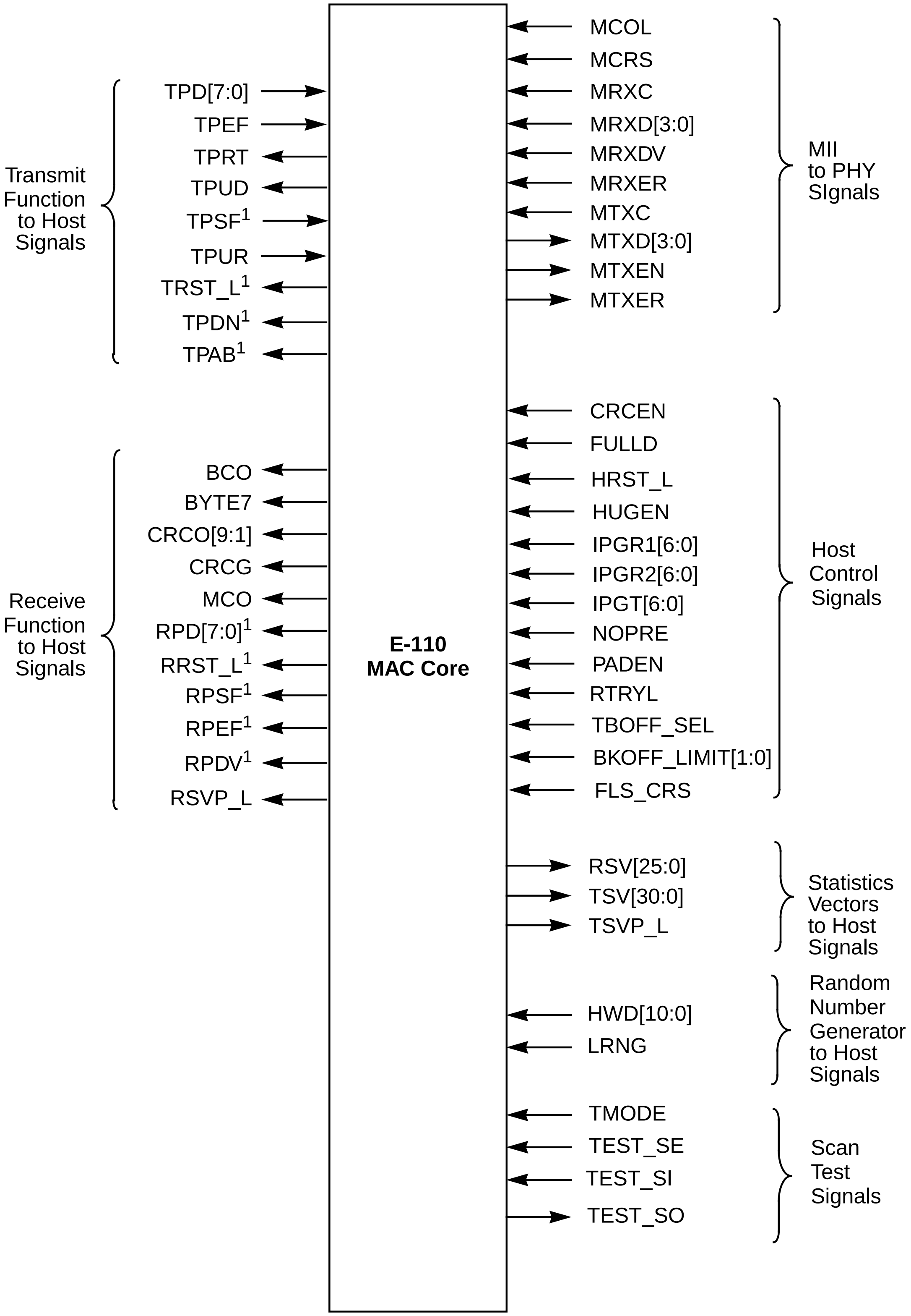
Please see the subsection entitled “Conventions Used in This Manual” on page viii of this manual for a description of how signals are named.

2.1 MAC Core Signals

The interface diagram for the E-110 MAC core is shown in Figure 2.1. The MAC signals fall into the following groups:

- Transmit Function
- Receive Function
- MAC Control Module
- MII to PHY
- Scan Test
- Host Control
- Statistics Vectors
- Random Number Generator

Figure 2.1 E-110 MAC Core Interface Diagram



[1] These signals are also connected to the optional MAC control module core, if it is used.

2.1.1 Transmit Function to Host Signals

Below is a list of the signals between the E-110 MAC transmit function and the host transmit buffer. All signals are active HIGH unless otherwise noted. All signals are synchronous with the rising edge of the MII PHY transmit clock (MTXC). Signal direction is with respect to the transmit function block. See the Section 3.2.1.1, “MAC Transmit Function,” on page 3-5 for further details on transmit buffer and transmit function interaction.

TPAB **Transmit Packet Abort** **Output**

When asserted, the TPAB signal indicates that the transmission was discontinued. TPAB remains asserted until the E-110 MAC function receives a request to transmit, which is indicated when the MAC control module asserts TPSF. When deasserted, TPAB indicates that the transmission was not aborted. The following circumstances cause the transmission to be halted:

- *Excess deferrals*, which occur when the media is busy longer than twice the maximum frame length (greater than 24,288[1] bits when the HUGEN signal is deasserted or greater than 524,288[2] bits when HUGEN is asserted)
- Late collision (use RTRYL to avoid aborting)
- Multiple collisions (greater than 15)
- Transmit underrun
- Larger than normal packet, which is 1518 bytes (see also see the HUGEN signal description)

The TPAB signal is also connected to the optional MAC control module core.

TPD[7:0] **Transmit Packet Data** **Input**

The TPD[7:0] signals are the transmit data bus. The host must hold the TPD[7:0] signals valid for exactly two MTXC clock cycles.

[1] 24,288 bits = 1518 bytes x 8 bits/byte x 2 (242.88 µs for 100-Mbit/s operation or 2.4288 ms for 10-Mbit/s operation)

[2] 524,288 bits = 32 Kbytes x 8 bits/byte x 2 (5242.88 µs for 100-Mbit/s operation or 52.43 ms for 10-Mbit/s operation)

TPDN **Transmit Packet Done** **Output**

When asserted, the TPDN signal indicates successful completion of the packet transmit process. The MAC keeps TPDN asserted until the MAC receives a fresh request to transmit, which is indicated when the MAC control module asserts the TPSF signal. The TPDN signal is also connected to the optional MAC control module core.

TPEF **Transmit Packet End of Frame** **Input**

The host asserts the TPEF signal to indicate the last byte of the transmit packet is available from the host. TPEF must be valid for two MTXC clock periods before it is deasserted.

TPRT **Transmit Packet Retry** **Output**

The MAC asserts the TPRT signal to indicate that the MAC function encountered at least one collision during a transmit attempt. The MAC asserts TPRT until the MAC receives a fresh request to transmit, which is indicated when the MAC control module asserts TPSF.

TPSF **Transmit Packet Start Of Frame** **Input**

The MAC control module, if implemented, asserts the TPSF signal to request the E-110 core to transmit a new packet. If the MAC control module is not implemented, the host must assert the TPSF signal.

This paragraph describes how the host must control the TPSF signal. The host interface asserts the TPSF signal to request the MAC to transmit a new packet. The host must keep the TPSF signal asserted for one transmit clock period after the E-110 core asserts the TPUD signal. The TPSF signal is synchronous to MTXC.

For a description of how the MAC control module controls the TPSF signal, see Section 2.1.3, “MAC Control Module Signals (optional)” on page 2-7.

TPUD **Transmit Packet Data Used** **Output**

The E-110 MAC function asserts the TPUD signal to indicate that the preamble has been transmitted. Every two clocks thereafter, the host must place the TPD[7:0] signals on the bus for the MAC. The MAC keeps TPUD asserted until the MAC accepts all the data bytes in the transmit packet from the host.

TPUR **Transmit Packet Data Underrun** **Input**

When the TPUR signal is asserted, the E-110 MAC function discontinues transmission. If the host is unable to supply transmit packet data bytes in a timely manner to the E-110 core (an underrun condition), the host must assert TPUR.

The host must assert TPUR for at least two MTXC clock cycles.

TRST_L **Transmit Reset** **Output**

Other modules in an ASIC can use the TRST_L signal as a host reset synchronized to the transmit clock (MTXC). Because the MTXC clock can be slow with respect to a host reset pulse, or even stopped, the host reset signal (HRST_L) is captured in the E-110 MAC transmit function. The transmit function asserts TRST_L asynchronously to MTXC when the HRST_L signal occurs and deasserts TRST_L synchronously on the positive transition of MTXC. Modules can use TRST_L to initialize transmit logic. The TRST_L signal is also connected to the optional MAC control module core.

2.1.2 Receive Function to Host Signals

Below is a list of the signals between the E-110 MAC receive function and the host receive logic. All signals are active HIGH unless otherwise noted. All signals are synchronous with the MRXC rising edge. Signal direction is with respect to the receive function block. See Section 3.2.1.2, “MAC Receive Function,” on page 3-13 for further details on receive buffer and receive function interaction via these signals.

BCO **Broadcast Out** **Output**

When asserted, the BCO signal indicates that the received packet is a *broadcast packet*. A broadcast packet has the destination address field set to all ones, which indicates it is being sent to all destinations on the network. When deasserted, BCO indicates that the received packet is not a broadcast packet. BYTE7 must be asserted for this output to be valid.

BYTE7 **Byte 7** **Output**

When asserted, the BYTE7 signal indicates that the BCO and MCO bits are valid. When deasserted, the BYTE7 signal indicates that the BCO and MCO signals are not valid.

CRCO[9:1] **CRC Out** **Output**

The CRCO[9:1] signals reflect the state of the receive function FCS register after the first six bytes of the receive packet have been received. When the destination address bits that are received in the frame contain a multicast address, the E-110 core uses its built-in FCS generator to compute a nine-bit polynomial (the nine MSBs of the 32-bit FCS generator) from the incoming address. The value of this polynomial can be used as an index into an external multicast filter hash table. External logic can decide to either accept or reject the incoming frame.

CRCG **CRC Good** **Output**

The CRCG signal, when asserted, indicates that the CRCO[9:1] signals are valid. The CRCG signal, when deasserted, indicates that the CRCO[9:1] signals are not valid.

MCO **Multicast Out** **Output**

When asserted, the MCO signal indicates that the received packet is a multicast packet (a packet that is being sent to selected stations on the network). When deasserted, MCO indicates that the received packet is not a multicast packet, which means that it could be an individual or broadcast packet. BYTE7 must be asserted for MCO to be valid.

RPD[7:0] **Receive Packet Data** **Output**

The RPD[7:0] signals are the receive data bus. The signals hold the received data byte for two MRXC clock cycles. The RPD[7:0] signals are also connected to the optional MAC control module core.

RPDV **Receive Packet Data Valid** **Output**

A packet transmission from the receive function to the host receive control logic (see Figure 1.6 on page 1-12) begins when the receive function asserts the RPSF and RPDV signal at the first byte of received packet data on RPD[7:0] after removing the preamble and SFD. For subsequent data bytes, the receive function asserts only the RPDV signal until the last byte, when it asserts both RPDV and RPEF. See Figure 4.1 on page 4-2 for receive packet timing. The RPDV signal is also connected to the optional MAC control module core.

RPEF **Receive Packet End of Frame** **Output**

The MAC asserts the RPEF signal for one MRXC clock cycle to indicate that the last byte of the receive packet is available to the MAC control module on RPD[7:0]. The RPEF signal is also connected to the optional MAC control module core.

RPSF **Receive Packet Start of Frame** **Output**

The MAC asserts the RPSF signal for one MRXC clock cycle to indicate that the first byte of a receive packet is available to the host on RPD[7:0]. The RPSF signal is also connected to the optional MAC control module core.

RRST_L **Receive Reset** **Output**

Other modules in an ASIC can use the RRST_L signal as a host reset synchronized to the receive clock (MRXC). Because the MRXC clock can be slow with respect to a host reset pulse, or even stopped, the host reset signal (HRST_L) is captured in the E-110 MAC receive function, which asserts RRST_L asynchronously to MRXC when HRST_L occurs and deasserts RRST_L synchronously on the positive transition of MRXC. The RRST_L signal is also connected to the optional MAC control module core.

RSVP_L **Receive Statistics Vector Pulse** **Output**

The RSVP_L signal is active LOW. When the MAC asserts RSVP_L, it indicates that the RSV[25:0] signals have been updated with a new receive statistics vector. When deasserted, it indicates that there has been no update. The RSVP_L signal is also connected to the optional MAC control module core.

2.1.3 MAC Control Module Signals (optional)

Below is a list of the signals that interface between the E-110 core and the optional MAC control module core. (See also Section 2.2, “MAC Control Module Signals,” on page 2-28.) All signals are active HIGH unless otherwise noted. Signal direction is with respect to the E-110 core.

RPD[7:0] **Receive Packet Data** **Output**

The RPD[7:0] signals are the receive data bus. The signals hold the received data byte for two MRXC clock cycles. The RPD[7:0] signals are also connected to the host.

RPDV **Receive Packet Data Valid** **Output**

A packet transmission from the receive function to the host receive control logic (see Figure 1.6 on page 1-12) begins when the receive function asserts the RPSF and RPDV signal at the first byte of received packet data on RPD[7:0] after removing the preamble and SFD. For subsequent data bytes, the receive function asserts only the RPDV signal until the last byte, when it asserts both RPDV and RPEF. See Figure 4.1 on page 4-2 for receive packet timing. The RPDV signal is also connected to the host.

RPEF **Receive Packet End of Frame** **Output**

The MAC asserts the RPEF signal for one MRXC clock cycle to indicate that the last byte of the receive packet is available to the MAC control module on RPD[7:0]. The RPEF signal is also connected to the host.

RPSF **Receive Packet Start of Frame** **Output**

The MAC asserts the RPSF signal for one MRXC clock cycle to indicate that the first byte of a receive packet is available to the host on RPD[7:0]. The RPSF signal is also connected to the host.

RRST_L **Receive Reset** **Output**

Other modules in an ASIC can use the RRST_L signal as a host reset synchronized to the receive clock (MRXC). Because the MRXC clock can be slow with respect to a host reset pulse, or even stopped, the host reset signal (HRST_L) is captured in the E-110 MAC receive function, which asserts RRST_L asynchronously to MRXC when HRST_L occurs and deasserts RRST_L synchronously on the positive transition of MRXC. The RRST_L signal is also connected to the host.

TPAB **Transmit Packet Abort** **Output**

When asserted, the TPAB signal indicates that the transmission was discontinued. TPAB remains asserted until the E-110 MAC function receives a request to transmit, which is indicated when the MAC control module asserts TPSF. When deasserted, TPAB indicates that the transmission was not aborted. The following circumstances cause the transmission to be halted:

- Excess deferrals, which occur when the media is busy longer than twice the maximum frame length (greater than 24,288[1] bits when the HUGEN signal is deasserted or greater than 524,288[2] bits when HUGEN is asserted)

- Late collision (use RTRYL to avoid aborting)
- Multiple collisions (greater than 15)
- Transmit underrun
- Larger than normal packet, which is 1518 bytes (see also see the HUGEN signal description)

The TPAB signal is also connected to the host.

TPDN **Transmit Packet Done** **Output**
When asserted, the TPDN signal indicates successful completion of the packet transmit process. The MAC keeps TPDN asserted until the MAC receives a fresh request to transmit, which is indicated when the MAC control module asserts the TPSF signal. The TPDN signal is also connected to the host.

TPSF **Transmit Packet Start Of Frame** **Input**
The MAC control module asserts the TPSF signal to request the E-110 Core to transmit a new packet. Once asserted, TPSF is kept asserted as long as the HST_TPSF signal from the host is asserted. On a data packet transmit request from the host (HST_TPSF asserted and CTLP deasserted), TPSF is asserted by the MAC control module only if PAUSE_ACTIVE is deasserted. The MAC control module does not enter the pause state because either the pause counter has counted down to zero or the counter is currently loaded with a zero value. If the PAUSE_ACTIVE signal is asserted, TPSF is not asserted until PAUSE_ACTIVE is deasserted.

If the FLOWCON_EN signal is deasserted, the MAC control module does not assert the TPSF signal for host control packet transmit requests. If the FLOWCON_EN signal is deasserted, the TPSF signal is asserted for host data packet transmit requests. When FLOWCON_EN is

[1] 24,288 bits = 1518 bytes x 8 bits/byte x 2 (242.88 µs for 100-Mbit/s operation or 2.4288 ms for 10-Mbit/s operation)

[2] 524,288 bits = 32 Kbytes x 8 bits/byte x 2 (5242.88 µs for 100-Mbit/s operation or 52.43 ms for 10-Mbit/s operation)

asserted, TPSF is asserted one clock after HST_TPSF is asserted for a data or a control packet transmit request. When FLOWCON_EN is deasserted, TPSF is asserted at the same clock as when HST_TPSF is asserted for a data packet request. The MAC control module does not interpret the transmit data in any way and transmit data is routed from the host to the E-110 MAC directly.

TRST_L **Transmit Reset** **Output**

Other modules in an ASIC can use the TRST_L signal as a host reset synchronized to the transmit clock (MTXC). Because the MTXC clock can be slow with respect to a host reset pulse, or even stopped, the host reset signal (HRST_L) is captured in the E-110 MAC transmit function. The transmit function asserts TRST_L asynchronously to MTXC when the HRST_L signal occurs and deasserts TRST_L synchronously on the positive transition of MTXC. Modules can use TRST_L to initialize transmit logic. The TRST_L signal is also connected to the host.

2.1.4 MII to PHY Signals

The signals that interface the MAC MII to the PHY (not including the MIIM interface) are listed below. All signals are active HIGH. Signal direction is with respect to the MAC MII function. See IEEE 802.3u standard documentation for further information. The standard IEEE MII signal name references are shown in brackets.

MCOL **Collision Detected** **Input**

The PHY asserts the MCOL [COL] signal asynchronously with minimum delay from the start of a collision on the media. The PHY deasserts MCOL to indicate no collision. MCOL is internally synchronized to the MTXC clock and in the worst case may take up to two clock cycles to be detected by the MAC transmit function.

MCRS **Carrier Sense** **Input**

The PHY asserts the MCRS [CRS] signal asynchronously with minimum propagation delay from the detection of a non-idle medium. The PHY deasserts MCRS when it detects an idle medium. The PHY also asserts MCRS with minimum propagation delay in response to MTXEN [TX_EN].

The PHY ensures that MCRS remains asserted throughout the duration of a collision condition.

MRXC **Receive Nibble or Symbol Clock** **Input**

The MRXC [RX_CLK] signal is a continuous clock that provides a timing reference for transfer of the MRXDV [RX_DV], MRXD[3:0] [RXD], and MRXER [RX_ER] signals from the PHY to the E-110 MAC. MRXC is an input from the PHY.

As long as the PHY is receiving a continuous signal from the medium and can recover the MRXC clock reference and supply MRXC, the PHY does not need to make a switch from the recovered clock reference to a nominal clock reference (for example, the MTXC [TX_CLK] continuous clock signal from the PHY). However, if the loss of a receive signal causes the PHY to lose the recovered clock reference, the PHY must source MXRC from a nominal clock reference.

If the PHY needs to make a switch from recovered clock to nominal clock, it deasserts the MRXDV signal. During the interval between MCRS and the assertion of MRXDV at the beginning of a frame, the PHY may extend a cycle of the MRXC clock by holding it in either the HIGH or LOW condition until the PHY has successfully locked onto the recovered clock. Successive cycles of MRXC must meet the duty cycle requirement.

While MRXDV is asserted, the PHY recovers MRXC from the received data. MRXC has a frequency equal to 25% of the data rate of the received signal and is synchronous to recovered data. The duty cycle is between 35% and 60% and is nominally 50%. For 100-Mbit/s operation, MRXC is nominally 25 MHz ±100 ppm, and the minimum MRXC HIGH and LOW times are 14 ns. For 10-Mbit/s operation, MRXC is nominally 2.5 MHz ± 100 ppm, and the minimum MRXC HIGH and LOW times are 140 ns.

When the MCRS signal is deasserted, the PHY provides MRXC at the PHY’s nominal clock rate (for example, the MTXC clock signal) and with nominal duty cycle. The minimum HIGH and LOW times are each 35% of the nominal MRXC period except for the transition between recovered clock frequency and nominal clock frequency, which occurs while MRXDV is deasserted. Following the transition from MRXDV asserted to MRXDV deasserted, the PHY can keep MRXC in either the HIGH or LOW condition to extend the MRXC clock by one cycle until the PHY is ready to provide MRXC from a nominal clock source. The maximum HIGH or LOW time for MRXC during this transition is two times the nominal clock period (a total of 80 ns for 25-MHz operation or 800 ns for 2.5-MHz operation).

MRXD[3:0] **Receive Nibble Data** **Input**

MRXD[3:0] [RXD] consists of four data signals that the PHY drives synchronously to the rising edge of the MRXC clock. For each MRXC period in which MRXDV is asserted, the PHY transfers four bits of data over the MRXD[3:0] signals to the E-110 MAC. MRXD[0] is the least significant bit. When MRXDV is deasserted, the MRXD[3:0] signals have no effect on the E-110 MAC.

For a frame to be correctly interpreted by the E-110 MAC, a completely formed SFD must be passed across the interface. A completely formed SFD is the octet 0b1010.1011, which follows seven identical octets of preamble (0b1010.1010).

MRXDV **Receive Data Valid** **Input**

The PHY asserts the MRXDV [RX_DV] signal to indicate that the PHY is presenting recovered and decoded nibbles on the MRXD[3:0] [RXD[3:0]] signals and that MRXC is synchronous to the recovered data. The PHY asserts MRXDV synchronously to the rising edge of MRXC. The PHY keeps MRXDV asserted from the first recovered nibble of the frame through the final recovered nibble and deasserts it prior to the first MRXC that follows the final nibble. MRXDV encompasses the frame, starting no later than the SFD and excluding any *end of frame delimiter*. The PHY may also assert MRXDV for transferring a validly decoded preamble.

MRXER **Receive Error** **Input**

The PHY asserts the MRXER [RX_ER] signal to indicate to the E-110 MAC that a media error (for example, a coding error) was detected somewhere in the frame presently being transferred from the PHY to the E-110 MAC. The PHY asserts MRXER synchronously to the rising edge of MRXC for one or more MRXC periods and then deasserts it. The PHY must assert MRXER for at least one MRXC clock period during the frame.

MTXC **Transmit Nibble or Symbol Clock** **Input**
The MTXC [TX_CLK] signal operates at a frequency of 25 or 2.5 MHz. MTXC is a continuous clock that provides a timing reference for transfer of the MTXEN [TX_EN], MTXD[3:0] [TXD[3:0]], and MTXER [TX_ER] signals from the E-110 MAC to the PHY. The PHY provides MTXC.

The MTXC frequency is 25% of the transmit data rate. A PHY operating at 100 Mbit/s provides an MTXC frequency of 25 MHz ± 100 ppm. A PHY operating at 10 Mbit/s provides a TX_CLK frequency of 2.5 MHz ± 100 ppm. The duty cycle of the TX_CLK signal is between 35% and 60%, inclusively.

MTXD[3:0] **Transmit Nibble Data** **Output**
The MTXD[3:0] signals are synchronous to the rising edge of MTXC.

MTXD[3:0] [TXD[3:0]] consists of four data signals that are synchronous to MTXC. For each MTXC period in which MTXEN is asserted, the PHY accepts the MTXD[3:0] signals for transmission. MTXD[0] is the least significant bit. When MTXEN is deasserted, the MTXD[3:0] signals have no effect on the PHY.

MTXEN **Transmit Enable** **Output**
The MTXEN [TX_EN] signal indicates that the E-110 MAC is presenting MTXD[3:0] nibbles on the MII for transmission. The MAC asserts MTXEN synchronously with the first nibble of the preamble. MTXEN remains asserted while all nibbles to be transmitted are presented to the MII. The MAC deasserts MTXEN prior to the first MTXC following the final nibble of a frame. MTXEN is synchronous to the rising edge of MTXC, and the PHY samples MTXEN synchronously.

MTXER **Transmit Coding Error** **Output**
The E-110 MAC function asserts the MTXER [TX_ER] signal synchronously to the rising edge of MTXC, and the PHY samples MTXER synchronously. When the MAC asserts MTXER for one MTXC clock period while MTXEN is also asserted, MTXER causes the PHY to transmit one or more symbols that are not part of the valid data or

delimiter set somewhere in the frame being transmitted to indicate that there has been a transmit coding error. If the MAC asserts the MTXER signal when a PHY is operating at 10 Mbit/s or when MTXEN is deasserted, the PHY must not allow the transmission of data to be affected.

2.1.5 Host Control Signals

The control lines from the host to the MAC core are listed below. All signals are active HIGH unless otherwise indicated. Signal direction is with respect to the MAC function. These signals are not synchronized on a packet boundary within the E-110 MAC. That is, any change in these signals is immediate, and changing the state of the signals during a packet can cause unexpected results.

BKOFF_LIMIT[1:0]

Backoff Limit **Input**

The BKOFF_LIMIT[1:0] signals determine the number of transmission attempts the E-110 MAC core makes after a collision and the integer number of slot times[1] the core waits before rescheduling a transmission attempt (during retries after a collision).

When a transmission attempt has terminated due to a collision, the MAC core retries until either the transmission is successful or the maximum number of attempts (16) have been made and all have terminated due to collisions.The MAC core waits an integer number of slot times (backoff) before each attempt to retransmit. The backoff delay, *r*, before each retransmission attempt is chosen as a uniformly distributed random integer in the range:

$$0 \leq r < 2^k$$

The variable *k* is the retry counter value and is calculated as follows:

$$k = min\langle n, 10\rangle$$

The variable *n* is the current number of retransmissions. The retry counter value is held to the lesser of the current number of retransmissions or the value 10.

[1] One slot time equals 512 bit times.

The following table illustrates the relationship between the backoff delay and the maximum retry counter value.

BKOFF_LIMIT		Maximum Backoff Delay in Slot Times	Retry Counter (k) Maximum Value
1	**0**		
0	0	0–1023	10
0	1	0–255	8
1	0	0–15	4
1	1	0–1	1

As an example, if the BKOFF_LIMIT[1:0] value is 0b10, the maximum retry counter value is 4. The E-110 MAC core retry counter value is zero before any collisions occur. When a collision occurs, the retry counter increments to one, and the backoff delay is set to a value between 0 and 1 slot times. When the backoff delay expires, another transmission attempt is made. If a collision occurs again, the retry counter goes to two and the backoff delay is set to between 0 and 3. Assuming a collision at each attempt, the process continues until the retry counter reaches four, at which time the backoff delay is set to between 0 and 15, and the retry counter rolls over to zero. If the next attempt also encounters a collision, the retry counter increments to one, and the backoff value is set to between 0 and 1. If collisions keep occurring, the MAC core keeps retrying for a total of 16 retransmission attempts and then stops retrying. The following table summarizes the transmission retry algorithm for all four cases of BKOFF_LIMIT[1:0].

Table 2.1 Transmission Retry Algorithm for BKOFF_LIMIT[1:0] (Maximum k = 10, 8, 4, 1)

	Attempt															
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Current k (Maximum = 10)	1	2	3	4	5	6	7	8	9	10	10	10	10	10	10	10
Backoff Delay	0–1	0–3	0–7	0–15	0–31	0–63	0–127	0–255	0–1023	0–1023	0–1023	0–1023	0–1023	0–1023	0–1023	0–1023
Current k (Maximum = 8)	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Backoff Delay	0–1	0–3	0–7	0–15	0–31	0–63	0–127	0–255	0–1	0–3	0–7	0–15	0–31	0–63	0–127	0–255
Current k (Maximum = 4)	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Backoff Delay	0–1	0–3	0–7	0–15	0–1	0–3	0–7	0–15	0–1	0–3	0–7	0–15	0–1	0–3	0–7	0–15
Current k (Maximum = 1)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Backoff Delay	0–1	0–1	0–1	0–1	0–1	0–1	0–1	0–1	0–1	0–1	0–1	0–1	0–1	0–1	0–1	0–1

The host must assert the BKOFF_LIMIT[1:0] signals synchronously with the rising edge of MTXC clock. The effect of these signals is not synchronized on a packet boundary within the E-110 core and changing the state of the signals during a packet transmission can cause unexpected results.

CRCEN **CRC Append Enable** **Input**

The host asserts the CRCEN signal to instruct the transmit function to append the FCS calculated by the MAC to the end of the transmitted data. When the host deasserts CRCEN, the E-110 core still calculates the FCS of the transmitted packet data, but allows the FCS that comes from the host to be transmitted with the packet. The E-110 core checks the host-generated FCS to see if it is valid except when the host asserts the NOPRE signal (see the “NOPRE No Preamble Input” signal description on page 2-20). If the FCS is not valid, the E-110 core asserts the FCS error signal (TSV21) in the Transmit Statistics Vector. CRCEN is synchronous with the rising edge of the MTXC clock and may be changed when the transmit engine is idle (either during reset or when no packet is being transmitted).

FLS_CRS **False Carrier Sense (Backpressure Control)** **Input**

The host may use the E-110 core FLS_CRS input pin to implement backpressure (see Section 1.4.6, “MAC Backpressure Feature,” on page 1-22). Backpressure makes the medium look busy to other stations on the network who wish to send data to the E-110 core. The host asserts the FLS_CRS input when a congestion threshold for the port’s input buffer is reached. The host should select the congestion threshold in such a way that it leaves enough room in the buffer for the frame in progress. When the FLS_CRS pin is asserted, the E-110 core waits until 44 bit times after the medium becomes inactive (CRS deasserted) and then starts sending out a false carrier data pattern (alternate ones and zeroes). The E-110 core continues sending this data pattern as long as the host continues to assert the FLS_CRS signal. If the E-110 core is already sending out normal packet data on the MII, assertion of the FLS_CRS pin only goes into effect after the current transmission is completed. If the E-110 core is already sending out a false carrier data pattern on the MII, any transmit requests from the host are kept pending until the FLS_CRS pin is deasserted. It is the responsibility of the host to disable the *jabber* timer if the E-110 core is being used in the 10BASE-T mode and if the FLS_CRS pin is asserted for more than 20 ms. The E-110 core ignores collisions during transmission of data while the FLS_CRS pin is asserted. Standard management information related to the Ethernet interface is not affected by the FLS_CRS signal.

FULLD **Full-Duplex Control** **Input**

When the host asserts the FULLD signal, the transmit function operates in full-duplex mode, does not defer to the CRS signal, and does not respond to the COL signal. When FULLD is deasserted, the transmit function operates in half-duplex mode. In half-duplex mode, the E-110 core defers to CRS and responds to COL. FULLD may be changed when the system is idle (either during reset or when both transmit and receive engines are idle). The host must assert FULLD synchronously to the rising edge of MTXC.

HRST_L **Host Reset** **Input**

The HRST_L signal is an active LOW signal that initializes the MAC function and the MIIM function when the host asserts it. When HRST_L is asserted, the MAC asserts the synchronized transmit and receive reset signals, TRST_L and RRST_L, which are inputs to the MAC transmit function and receive function, respectively. Other modules in an ASIC may use these signals for initialization. Both TRST_L and RRST_L are asserted LOW asynchronously when HRST_L occurs and are deasserted synchronously with their respective clocks (TRST_L with MTXC and RRST_L with MRXC).

The MAC assumes that HRST_L is asynchronous to all clocks. The minimum reset width is 400 ns for the 100-Mbit/s mode of operation and 4000 ns for the 10-Mbit/s mode.

HUGEN **Huge Packet Enable** **Input**

A packet consists of the following fields: Destination Address (six bytes), Source Address (six bytes), Data Length (two bytes), Data (the 802.3 standard stipulates a maximum of 1500 bytes), and Frame Check Sequence (four bytes). A packet can therefore be up to 1518 bytes and meet the 802.3 standard requirements.

However, as applications have evolved over a period of time, it is required that MAC controllers have the flexibility to transmit and receive packets of various sizes (in addition to supporting the conventional maximum packet sizes specified by the 802.3 standard).

When asserted, the HUGEN signal indicates to the transmit function in the E-110 core to allow transmission of packets up to 32 Kbytes. The E-110 core truncates any data packet larger than 32 Kbytes. Assertion of the HUGEN signal also instructs the receive function to receive packets up to 32 Kbytes. When HUGEN is asserted, the excess deferral value is changed to 131,072 nibble clocks (524,288 bit times, or two times the maximum packet size bit times). TSV[15:0] and RSV[15:0] reflect the count of number of bytes transmitted or received. When HUGEN is deasserted, the maximum size packet supported by the E-110 core is 1518 bytes. (Note that excess deferral is different this time as the packet size changes.)

When the MAC transmits a packet, the MAC truncates any data beyond 1536 bytes (when HUGEN is deasserted) or 32 Kbytes (when HUGEN is asserted) and asserts the TSV26 signal (see Section 2.1.6, "Statistics Vector to Host Signals," on page 2-21), which indicates that the packet was aborted because of excessive length. Note that the MAC does not begin truncating the data immediately after 1518 bytes.

When the MAC receives a packet longer than 1518 bytes (with HUGEN deasserted), or 32 Kbytes (with HUGEN asserted), the MAC asserts the RSV23 signal (see Section 2.1.6, "Statistics Vector to Host Signals"), which indicates that a bad packet has been received. However, the MAC continues to accept bytes until 1536 bytes or 32 Kbytes, beyond which it stops accepting bytes.

IPGR1[6:0] **Non Back-to-Back IPG (Part 1)** **Input**

The IPGR1[6:0] signals contain the programmable transmit interpacket gap (IPG) for non back-to-back transmits, Part 1. Non back-to-back transmits allow a receive to occur between transmits. The MAC core samples the IPGR1[6:0] signals synchronously with the rising edge of the MTXC clock during reset. For a detailed discussion of the IPGR1[6:0] signals, see "IPG Half-Duplex Operation" on page 3-6.

IPGR2[6:0] **Non Back-to-Back IPG (Part 2)** **Input**

The IPGR2[6:0] signals contain the programmable transmit IPG for non back-to-back transmits, Part 2. The MAC core samples the IPGR2[6:0] signals synchronously with the rising edge of the MTXC clock during reset. For a detailed discussion of IPGR2[6:0], see "IPG Half-Duplex Operation" and "IPG Full-Duplex Operation" on page 3-6.

IPGT[6:0] **Back-to-Back IPG** **Input**

The IPGT[6:0] signals contain the programmable transmit IPG for back-to-back transmits. Nominal operation requires that IPGT[6:0] have a value of 18. For a detailed discussion of IPGT[6:0], see "IPG Half-Duplex Operation" on page 3-6.

The MAC core samples the IPGT[6:0] signals synchronously with the rising edge of the MTXC clock during reset. For a detailed discussion of IPGT[6:0], see "IPG

Half-Duplex Operation” and “IPG Full-Duplex Operation” on page 3-8.

NOPRE **No Preamble** **Input**

When asserted, the NOPRE signal instructs the transmit function to disable the addition of the preamble. Padding and FCS appending are also disabled when NOPRE is asserted. NOPRE is synchronous with the rising edge of the MTXC clock and may be changed when the transmit engine is idle (either during reset or when no packet is being transmitted). When deasserted, the NOPRE signal allows the transmit function to add the preamble, perform byte padding, and do FCS appending.

PADEN **Pad Enable** **Input**

The PADEN signal, when asserted, instructs the transmit function to pad packets of fewer than 60 bytes with a sufficient number of bytes of zero such that the minimum packet size (64 bytes, including data plus an FCS of 4 bytes) specified by IEEE 802.3 is maintained. When deasserted, PADEN disables the padding of packets. For padding to take place, both PADEN and CRCEN must be asserted. PADEN is synchronous with the rising edge of the MTXC clock and may be asserted or deasserted when the transmit engine is idle (either during reset or when no packet is being transmitted).

RTRYL **Retry Late Collision** **Input**

The RTRYL signal, when asserted, instructs the transmit function to retry transmission after a late collision (a collision that occurs after 512 bit times into the packet) instead of aborting it. When deasserted, RETRYL instructs the E-110 core not to retry a transmission after a late collision, to reset the retry counter, and to abort the transmit attempt.

For normal collisions (a collision that occurs fewer than 512 bit times into the packet) the transmit function retries the transmission up to 15 times before aborting. If late collisions occur frequently, it may indicate that there is a problem with the network cabling or equipment. The TSV29 signal (see Section 2.1.6, “Statistics Vector to Host Signals,” on page 2-21) in the Transmit Statistics Vector, when asserted, indicates the transmission was dropped because of a late collision.

RTRYL is synchronous with the rising edge of MTXC and may be asserted or deasserted when the transmit engine is idle (either during reset or when no packet is being transmitted).

TBOFF_SEL **Stop Backoff** **Input**

The TBOFF_SEL input signal controls whether or not the E-110 core uses the binary exponential backoff algorithm during collisions. If the TBOFF_SEL signal is asserted, the E-110 core transmits a *jam* pattern after a collision and tries to retransmit after timing out for the IPG value. In this case, the binary exponential backoff algorithm is not used.

If the TBOFF_SEL signal is deasserted, the E-110 core implements the binary exponential backoff algorithm. The E-110 core stops sending packet data, sends a jam pattern, and tries to transmit again. The amount of time the E-110 core waits between successive retries is 512 bit times[1] x R, where R is a random integer between 0 and $2^k - 1$. K is the retry counter value, which can range from 1 to 10.

The host must change the TBOFF_SEL signal synchronously with the rising edge of MTXC clock. The effect of this signal is not synchronized on a packet boundary within the E-110 core, so changing the state of the signal during a packet transmission can cause unexpected results.

2.1.6 Statistics Vector to Host Signals

The statistics vector signal lines from the MAC are listed below. All signals are active HIGH unless otherwise indicated. Signal direction is with respect to the MAC.

RSV[25:0] **Receive Statistics Vector** **Output**

The RSV[25:0] signals contain the receive statistics vector and are updated on the falling edge of RSVP_L. The statistics vector is issued at the end of any *minimally qualified receive event*. A minimally qualified receive event occurs when the MAC receives at least one nibble of data beyond a valid preamble and SFD.

[1] 512 bit times equals one slot time.

The RSV[25:0] signals are stable until the subsequent RSVP_L pulse. The condition associated with each signal is valid when the signal is HIGH.

The receive statistics provided by the MAC function can be used for RMON (remote monitoring) and SNMP (simple network management protocol). However, the MAC does not collect the statistics specifically mentioned in the RMON and SNMP specifications. The MAC provides basic per packet information that can be collected by an application built on top of the MAC. The collected information can then be used for RMON and SNMP.

The signals in RSV[25:0] have the following functions:

RSV25 Carrier event previously seen. When the MAC asserts the RSV25 signal, it indicates that at some time since the last receive vector a carrier was detected, noted, and reported with this vector. The carrier event is not associated with this packet. A carrier event is defined as activity on the receive channel that does not result in a *packet receive attempt*. An example would be receiving a preamble but no SFD, or receiving more than seven octets of preamble. A carrier event can occur in either full- or half-duplex modes. If RSV25 is deasserted, a carrier was not detected.

RSV24 Good packet received. For 100-Mbit/s operation, a good packet has no 4B/5B receive code-group violations, no dribble nibbles, a valid FCS, and a proper packet length (at least 64 bytes but not more than 1518 bytes or 32 Kbytes). 4B/5B code-group violations include the following: 0b00100, 0b00000, 0b00001, 0b00010, 0b00011, 0b00101, 0b00110, 0b01000, 0b01100, 0b10000, and 0b11001.

For 10-Mbit/s operation, a good packet has no *dribble nibbles*, a valid FCS, and a proper packet length (at least 64 bytes but not more than 1518 bytes or 32 Kbytes).

RSV23 Bad packet received. For 100-Mbit/s operation, a bad packet contains at least one of the following problems: 4B/5B code violations, dribble nibbles, bad FCS, less than 64 bytes (short packet), or more than 1518 bytes or 32 Kbytes (long packet).

For 10-Mbit/s operation, a bad packet contains at least one of the following problems: Dribble nibbles, a bad FCS, less than 64 bytes (short packet), or more than 1518 bytes or 32 Kbytes (long packet).

RSV22 *Long event* previously seen. When the MAC asserts the RSV22 signal, it indicates that at some time since the last receive vector a long event was detected, noted, and reported with this vector. The long event is not associated with this packet. A long event is activity on the network in excess of 50,000 bit times. A long event is not detected in full-duplex mode. If RSV22 is deasserted, it indicates that a long event was not seen.

RSV21 *Invalid preamble* content (not 55H) or code (0x50555) seen in the last reception.

RSV20 Broadcast packet received (all ones in the destination address field).

RSV19 Multicast packet received. (The first bit in the destination address field, which is also the least significant bit, is equal to a one—the least significant bit is transmitted first.)

RSV18 FCS error detected in the received packet. The FCS bytes in the received packet do not match those the MAC calculates at the destination while the packet is being received.

RSV17 Dribble nibble seen in the received packet. Each byte consists of two nibbles, so the number of received nibbles should always be even. If there is an odd number of nibbles, the last nibble is the dribble nibble.

RSV16 Receive code violation detected. There is a 4B/5B code violation. See the RSV24 signal description for a definition of code-group violations. The PHY asserts MRXER whenever a receive code-group violation occurs.

RSV15 (msb) Packet length bit 15. If the host asserts the HUGEN signal, the Packet Length (Source Address, Destination Address, Data Length, Data, and FCS fields) as indicated by the Packet Length bits can be up to 32 Kbytes. Bits 15 through 0 form a 16-bit binary counter, with the least significant bit (RSV0) = 1 byte.

RSV14 Packet length bit 14.

RSV13 Packet length bit 13.

RSV12 Packet length bit 12.

RSV11 Packet length bit 11.

RSV10 Packet length bit 10.

RSV9 Packet length bit 9.

RSV8 Packet length bit 8.

RSV7 Packet length bit 7.

RSV6 Packet length bit 6.

RSV5 Packet length bit 5.

RSV4 Packet length bit 4.

RSV3 Packet length bit 3.

RSV2 Packet length bit 2.

RSV1 Packet length bit 1.

RSV0 (lsb) Packet length bit 0.

TSV[30:0] **Transmit Statistics Vector** **Output**

The TSV[30:0] signals contain the transmit statistics information. The MAC function updates the TSV[30:0] signals on the falling edge of the TSVP_L signal. The MAC issues the statistics vector at the end of the final or only attempt to transmit each packet, whether the packet is transmitted or not. The TSV[30:0] signals are stable until the subsequent TSVP_L pulse. The condition associated with each signal is valid when the signal is HIGH.

The MAC function provides transmit statistics that can be used for RMON and SNMP. However, the MAC does not collect the statistics specifically mentioned in the RMON and SNMP specifications. The MAC provides basic per packet information that can be collected by an application built on top of the MAC. The collected information can then be used for RMON and SNMP.

The signals in TSV[30:0] have the following functions:

TSV30 Transmit canceled because of excess deferral. Excess deferrals occur when the network is constantly busy (greater than 24,288 bit times when HUGEN is deasserted or greater than 524,288 bit times when HUGEN is asserted).

TSV29 Transmit dropped because of late collision. A late collision is one that occurs greater than 512 bit times into packet transmission.

TSV28 Transmit dropped because of excessive collisions (15 transmit retries).

TSV27 Transmit aborted because of underrun. If the host is unable to supply transmit packet data bytes in a timely manner to the E-110 core an underrun condition exists.

TSV26 Transmit aborted because of excessive length. The transmission is aborted if the packet exceeds 1518 bytes with the HUGEN signal LOW, or 32 Kbytes with the HUGEN signal HIGH.

TSV25 Packet transmitted successfully.

TSV24 Packet deferred on transmission attempt. A packet is deferred when the network is busy.

TSV23 Broadcast packet transmitted or attempted.

TSV22 Multicast packet transmitted or attempted.

TSV21 FCS error seen on transmission attempt. When the host deasserts CRCEN, indicating that the host provides the FCS field in transmitted packets, the MAC verifies the host FCS against its internally computed FCS. The E-110 core asserts the TSV21 signal to indicate a mismatch in the two FCSs. If the host asserts the CRCEN signal, the MAC both calculates and provides the FCS in transmitted packets. In this case, the MAC does not assert the TSV21 signal.

TSV20 Late collision (a collision that occurs more than 512 bits times into the packet) seen on transmission attempt.

TSV19 (msb) Collision count bit 3. The collision count can range from 0 to 15 for a packet ultimately transmitted, but can never be 16. After 15 retries, the packet is dropped because of excessive collisions. Bits 3 through 0 form a 4-bit binary counter, with the least significant bit = 1 collision.

TSV18 Collision count bit 2.

TSV17 Collision count bit 1.

TSV16 (lsb) Collision count bit 0.

TSV15 (msb) Packet length bit 15. The Packet Length bits indicate the length of the packet in bytes. The packet includes the Source Address, Destination Address, Data Length, Data, and FCS fields. Bits 15 through 0 form a 16-bit binary counter, with the least significant bit (TSV0) = 1 byte.

TSV14 Packet length bit 14.

TSV13 Packet length bit 13.

TSV12 Packet length bit 12.

TSV11 Packet length bit 11.

TSV10 Packet length bit 10.

TSV9 Packet length bit 9.

TSV8 Packet length bit 8.

TSV7 Packet length bit 7.

TSV6 Packet length bit 6.

TSV5 Packet length bit 5.

TSV4 Packet length bit 4.

TSV3 Packet length bit 3.

TSV2 Packet length bit 2.

TSV1 Packet length bit 1.

TSV0 (lsb) Packet length bit 0.

TSVP_L **Transmit Statistics Vector Pulse** **Output**

The TSVP_L signal is active LOW. When asserted, it indicates that the TSV[30:0] signals have been updated with a new transmit statistics vector. When deasserted, it indicates that there has been no update.

2.1.7 Random Number Generator to Host Signals

Below is a list of the signals that interface the E-110 MAC random number generator function to the host. All signals are active HIGH unless otherwise noted. Signal direction is with respect to the random number generator.

HWD[10:0] **Host Write Data [10:0]** **Input**
The HWD[10:0] signals contain the value of the random number that the host loads into the MAC's Linear Feedback Shift Register (LFSR) to generate the random number sequence used in collision backoff timing. HWD[10:0] must remain stable for two MTXC cycles after LRNG is asserted.

LRNG **Load Random Number Generator** **Input**
The host asserts the LRNG signal to indicate that the HWD[10:0] signals are valid and the MAC function should latch them. When the host deasserts LRNG, the HWD[10:0] signals are not valid. LRNG must be synchronous to the MTXC clock and at least one MTXC clock cycle wide, which is 40 ns for 100-MHz operation and 400 ns for 10-MHz operation (see the description of the MTXC signal on page 2-13). The HWD[10:0] signals must be stable when the host pulses LRNG.

2.1.8 Scan Test Signals

The MAC design uses full scan test methodology, which consists of a single scan chain and appropriate scan signals. The scan signals are described below.

TMODE **Test Mode** **Input**
When the TMODE signal is asserted, the MAC switches to test mode. When TMODE is deasserted, the MAC resumes normal functional operation.

TEST_SE **Scan Enable** **Input**
When the TEST_SE signal is asserted, all of the registers in the MAC accept data from their test inputs instead of from their normal inputs. When TEST_SE is asserted, the

scan data can be shifted in on the TEST_SI pin. TEST_SE must be kept deasserted during normal operation.

TEST_SI **Scan Input** **Input**
The TEST_SI signal is the serial scan chain input pin. When the TMODE and TEST_SE signals are asserted, test data can be presented to the MAC on the TEST_SI pin.

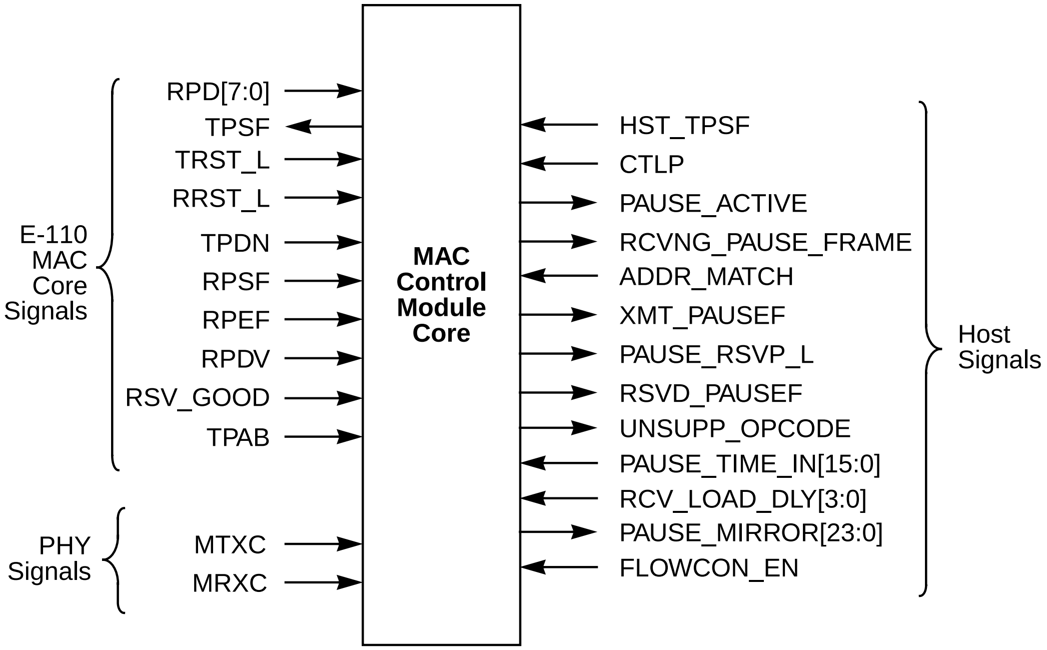
TEST_SO **Scan Output** **Output**
The TEST_SO signal is the serial scan chain output pin. The data presented on TEST_SO is the output data that results from the input stimulus on TEST_SI.

2.2 MAC Control Module Signals

The interface diagram for the optional MAC control module core is shown in Figure 2.2. The signals fall into the following groups:

- E-110 MAC core signals
- PHY
- Host Signals

Figure 2.2 MAC Control Module Core Interface Diagram



2.2.1 MAC Control Module to E-110 Core Signals

The signals that connect the optional MAC control module to the E-110 core are listed below (see also Section 2.1.3, “MAC Control Module Signals (optional),” on page 2-7). All signals are active HIGH unless otherwise indicated. Signal direction is with respect to the MAC control module.

RPD[7:0] **Receive Packet Data** **Input**
The RPD[7:0] signals are the receive data bus. The signals hold the received data byte for two MRXC clock cycles.

RPDV **Receive Packet Data Valid** **Input**
A packet transmission from the receive function begins when the receive function asserts the RPSF and RPDV signal at the first byte of received packet data on RPD[7:0] after removing the preamble and SFD. For subsequent data bytes, the receive function asserts only the RPDV signal until the last byte, when it asserts both RPDV and RPEF.

RPEF **Receive Packet End of Frame** **Input**
The MAC asserts the RPEF signal for one MRXC clock cycle to indicate that the last byte of the receive packet is available to the MAC control module on RPD[7:0].

RPSF **Receive Packet Start of Frame** **Input**
The MAC asserts the RPSF signal for one MRXC clock cycle to indicate that the first byte of a receive packet is available to the MAC control module on RPD[7:0].

RRST_L **Receive Reset** **Input**
Other modules in an ASIC can use the RRST_L signal as a host reset synchronized to the receive clock (MRXC). Because the MRXC clock can be slow with respect to a host reset pulse, or even stopped, the host reset signal (HRST_L) is captured in the E-110 MAC receive function, which asserts RRST_L asynchronously to MRXC when HRST_L occurs and deasserts RRST_L synchronously on the positive transition of MRXC.

RSV_GOOD **Receive Statistics Vector Good** **Input**
At the end of a receive frame, the MAC updates the Receive Statistics Vector (RSV[25:0]). The MAC asserts

the RSV_GOOD signal, which reflects the RSV24 bit, if the received frame has no errors. The MAC deasserts the signal if the received frame contains errors.

TPAB **Transmit Packet Abort** **Input**

When asserted, the TPAB signal indicates that the transmission was discontinued. TPAB remains asserted until the E-110 MAC function receives a request to transmit, which is indicated when the MAC control module asserts TPSF. When deasserted, TPAB indicates that the transmission was not aborted. The following circumstances cause the transmission to be halted:

- Excess deferrals, which occur when the media is busy longer than twice the maximum frame length (greater than 24,288[1] bits when the HUGEN signal is deasserted or greater than 524,288[2] bits when HUGEN is asserted)
- Late collision (use RTRYL to avoid aborting)
- Multiple collisions (greater than 15)
- Transmit underrun
- Larger than normal packet, which is 1518 bytes (see also see the HUGEN signal description)

TPDN **Transmit Packet Done** **Input**

The E-110 core asserts the TPDN signal after successful transmission of a packet. TPDN is asserted until a new TPSF signal is asserted by the MAC control module. When the E-110 asserts TPDN, the MAC control module goes back to the idle state. The TPDN signal is synchronous to MTXC.

TPSF **Transmit Packet Start Of Frame** **Output**

The MAC control module asserts the TPSF signal to request the E-110 core to transmit a new packet. Once asserted, TPSF is kept asserted as long as the HST_TPSF signal from the host is asserted. On a data packet transmit request from the host (HST_TPSF asserted and CTLP deasserted), TPSF is asserted by the

[1] 24,288 bits = 1518 bytes x 8 bits/byte x 2 (242.88 μs for 100-Mbit/s operation or 2.4288 ms for 10-Mbit/s operation)

[2] 524,288 bits = 32 Kbytes x 8 bits/byte x 2 (5242.88 μs for 100-Mbit/s operation or 52.43 ms for 10-Mbit/s operation)

MAC control module only if PAUSE_ACTIVE is deasserted. The MAC control module does not enter the pause state because either the pause counter has counted down to zero or the counter is currently loaded with a zero value. If the PAUSE_ACTIVE signal is asserted, TPSF is not generated until PAUSE_ACTIVE is deasserted.

If the FLOWCON_EN signal is deasserted, the MAC control module does not assert the TPSF signal for host control packet transmit requests. If the FLOWCON_EN signal is deasserted, the TPSF signal is asserted for host data packet transmit requests. When FLOWCON_EN is asserted, TPSF is asserted one clock after HST_TPSF is asserted for a data or a control packet transmit request. When FLOWCON_EN is deasserted, TPSF is asserted at the same clock as when HST_TPSF is asserted for a data packet request. The MAC control module does not interpret the transmit data in any way and transmit data is routed from the host to the E-110 MAC directly.

TRST_L **Transmit Reset** **Input**

Other modules in an ASIC can use the TRST_L signal as a host reset synchronized to the transmit clock (MTXC). Because the MTXC clock can be slow with respect to a host reset pulse, or even stopped, the host reset signal (HRST_L) is captured in the E-110 MAC transmit function. The transmit function asserts TRST_L asynchronously to MTXC when the HRST_L signal occurs and deasserts TRST_L synchronously on the positive transition of MTXC. Modules can use TRST_L to initialize transmit logic.

2.2.2 MAC Control Module to PHY Signals

The signals between the PHY and the MAC control module are listed below. Signal direction is with respect to the MAC control module.

MRXC **Receive Nibble or Symbol Clock** **Input**

The MRXC clock is a continuous clock signal that operates at 25 MHz or 2.5 MHz and provides a timing reference for data transfers from the E-110 core to the MAC control module.

MTXC **Transmit Nibble or Symbol Clock** **Input**

The MTXC clock is a continuous clock signal that operates at 25 MHz or 2.5 MHz and provides a timing reference for data transfers from the MAC control module to the E-110 core.

2.2.3 MAC Control Module to Host Signals

The signals that connect the MAC control module to the host are listed below. All signals are active HIGH unless otherwise indicated. Signal direction is with respect to the MAC control module.

ADDR_MATCH

Unicast Address Match **Input**

When asserted the ADDR_MATCH signal indicates to the MAC control module that the destination address of a pause frame matched the unicast address of the E-110 MAC core. The ADDR_MATCH signal should be asserted at least two clocks before the type field is presented on the host receive interface. This signal is synchronous to MRXC.

CTLP **Host Control Packet Transmit Request** **Input**

When asserted, the CTLP signal informs the MAC control module to treat a host packet transmit request (HST_TPSF asserted) as a pause frame transmit request. When the CTLP signal is deasserted, the MAC control module treats the host packet transmit request as a data packet request. The CTLP signal is synchronous to MTXC.

FLOWCON_EN

Flow Control Enable **Input**

When this pin is deasserted:

No action is taken to load the pause time and the PAUSE_TIMER is reset.

The Receive State Machine stays in the idle state.

If the host requests to transmit a control packet, the request is ignored.

If the host requests to transmit a data packet, the TPSF signal is asserted at the same clock as the HST_TPSF signal and the MAC control module is transparent to the host and E-110 MAC core.

The host must assert the FLOWCON_EN signal synchronously with the rising edge of the MTXC clock before transmission begins. The FLOWCON_EN signal is static and must not be used to reconfigure the logic on the fly. The effect of this signal is not synchronized on a packet boundary within the E-110 control module and changing the state of this signal during a packet transmission can cause unexpected results.

HST_TPSF **Host Packet Transmit Request** **Input**
The host interface asserts the HST_TPSF signal to request the MAC control module to transmit a new packet. The host must keep the HST_TPSF signal asserted for one transmit clock period after the E-110 core asserts the TPUD signal. The HST_TPSF signal is synchronous to MTXC.

PAUSE_ACTIVE
Pause in Progress **Output**
When asserted the PAUSE_ACTIVE signal indicates to the host that the MAC control module is in the pause state and cannot transmit any data frames. The PAUSE_ACTIVE signal is synchronous to MRXC.

PAUSE_MIRROR[23:0]
Mirror Counters Value **Output**
The PAUSE_MIRROR[23:0] signals contain the counter bits that reflect the value of the receiver's pause timer. The PAUSE_MIRROR[23:0] signals are valid after the MAC control module core asserts the XMT_PAUSEF signal.

PAUSE_RSVP_L
Comprehensive Receive Statistics Vector Pulse **Output**
On the falling edge of the PAUSE_RSVP_L signal, the RSVD_PAUSEF and UNSUPP_OPCODE signals are valid.

RCV_LOAD_DLY[3:0]
Receiver Delay in Loading Transmitted Pause **Input**
The 4-bit value contained in the RCV_LOAD_DLY[3:0] signals are in terms of slot times and accounts for the total time a receiver takes to load the transmitted pause value into its own pause counter. The 4-bit value is

loaded into the mirror counter when the HST_TPSF and CTLP signals are asserted. The MAC control module adds the 4-bit value to the PAUSE_TIME_IN[15:0] value in order to synchronize with a receiver's pause timer.

The RCV_LOAD_DLY[3:0] signals must be stable before the CTLP signal is asserted.

RCVNG_PAUSE_FRAME

Receiving Pause Frame **Output**

This signal is asserted by the MAC control module to indicate to the host that a pause frame has been received. This helps host to filter pause frames, if required. The MAC control module asserts the RCVNG_PAUSE_FRAME signal two clocks after the opcode field is presented on the receive data bus during valid reception of a pause frame (provided that the address, type, and opcode field match flags are set). Once asserted, this signal is held asserted until the completion of valid reception of a pause frame. This signal is synchronous to MRXC.

RSVD_PAUSEF

Received Pause Frame **Output**

The RSVD_PAUSEF signal, when asserted, indicates that the current packet received is a valid pause frame. The MAC control module core deasserts the signal if a data packet is received. This signal is valid on the falling edge of PAUSE_RSVP_L.

PAUSE_TIME_IN[15:0]

Transmitted Pause Time **Input**

The PAUSE_TIME_IN[15:0] signals constitute the 16-bit pause time value (in slot times) sent in the transmit control packet. The MAC control module loads the 16-bit value into its mirror counter when the HST_TPSF and CTLP signals are asserted. The PAUSE_TIME_IN[15:0] signals must be stable before the CTLP signal is asserted.

UNSUPP_OPCODE

Unsupported Opcode **Output**

The MAC control module asserts the UNSUPP_OPCODE signal if any opcode other than the pause opcode is received in a valid control frame. This signal is valid on the falling edge of PAUSE_RSVP_L.

XMT_PAUSEF

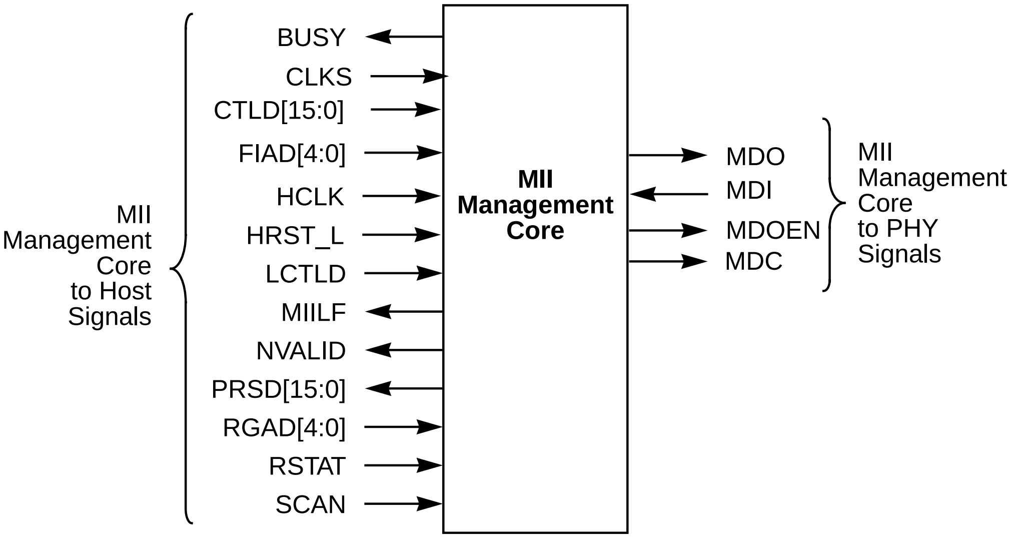
Transmitted Pause Frame **Output**

The XMT_PAUSEF vector signal is asserted after the transmission of a pause frame. It is deasserted if a data packet is transmitted. This signal is valid on the falling edge of the TSVP_L signal, which is asserted by the E-110 core.

2.3 MIIM Core Signals

The interface diagram for the optional MIIM core is shown in Figure 2.3.

Figure 2.3 MIIM Core Interface Diagram



2.3.1 MIIM Core to Host Signals

The signals that connect the MIIM core to the host are listed below. All signals are active HIGH unless otherwise indicated. Signal direction is with respect to the MIIM core.

BUSY **Busy** **Output**

The MIIM core asserts the BUSY signal to indicate to the host that a read, write, or scan operation is in progress. BUSY is deasserted when no MII operation is in progress.

If a read, write, or scan command is issued while the BUSY signal is asserted, the results are unpredictable.

CLKS **Clock Select** **Input**

The host asserts the CLKS signal to indicate that the HCLK signal is 33 MHz. When deasserted, it informs the MIIM that HCLK is 25 MHz. The MIIM uses CLKS to determine how to divide down HCLK to create the MDC signal (see the “HRST_L Host Reset Input” signal description on page 2-18). The host must assert CLKS synchronously to the rising edge of HCLK, because CLKS is used in the clock divider circuit, which runs on HCLK.

CTLD[15:0] **Control Data [15:0]** **Input**

The host places data to be written to the MII PHY on the CTLD[15:0] control data bus. After the host asserts the LCTLD signal, the host must observe the BUSY signal and keep the CTLD[15:0] signals valid until the MIIM core deasserts the BUSY signal. The host uses the CTLD[15:0] signals to write the Control register of a PHY. For a definition of the Control register, see the subsection entitled “Write PHY Control Register” on page 3-19.

FIAD[4:0] **PHY Address [4:0]** **Input**

The FIAD[4:0] signals contain the address of the PHY the host has selected for reading or writing. The host must not change the FIAD[4:0] signals while the MIIM core is asserting the BUSY signal; otherwise, the MIIM core may read the FIAD[4:0] signals incorrectly. The MIIM core can support up to 32 PHYs.

HCLK **Host Clock** **Input**

The host clock is either a 33-MHz or 25-MHz clock. The CLKS signal indicates the HCLK frequency to the MIIM. The host clock generates the MIIM Data Clock (MDC), which has minimum HIGH and LOW times of 200 ns each. A 33-MHz HCLK is divided by 14 to generate a 2.35 MHz MDC and a 25-MHz HCLK is divided by 10 to generate a 2.5 MHz clock. To meet the minimum HIGH and LOW times, the maximum frequency of MDC is 2.5 MHz.

HRST_L **Host Reset** **Input**

The HRST_L signal is an active LOW signal that initializes the MAC function and the MIIM function when the host asserts it.

LCTLD **Load Control Data** **Input**

The host asserts the LCTLD signal to indicate that the MIIM core may begin an MII write sequence. When the host asserts LCTLD, the host must ensure that the CTLD[15:0], FIAD[4:0], and RGAD[4:0] signals are valid and must not change them during the entire MII write operation. The MIIM core requires that the LCTLD signal be at least one HCLK pulse wide, which is 30.3 ns for a 33-MHz HCLK or 40 ns for a 25-MHz HCLK (see the HCLK signal description on page 2-18). LCTLD must be asserted and deasserted synchronously with the rising edge of HCLK.

MIILF **MII Link Failure** **Output**

When the host uses the MIIM core to read a PHY's status register, the MIILF output signal from the MIIM core reflects the state of the Link Status bit (bit 2) in the PHY's MII status register (see Table 3.3 on page 3-21); that is, if the Link Status bit is one, MIILF is asserted, indicating a *link pass*. If the bit is zero, MIILF is deasserted.

The MIIM core updates the MIILF signal whenever a status read operation or a scan cycle has accessed the PHY status register (register address 0x01). The MIIM core does not interpret the Link Status bit in any way.

During a read operation, MIILF is valid after BUSY is deasserted. During a scan operation, MIILF is valid after NVALID is deasserted.

NVALID **Invalid** **Output**

During a scan operation, the NVALID signal indicates the validity of the MIILF and PRSD[15:0] signals. When NVALID is asserted, it indicates that MIILF and PRSD[15:0] are invalid. When NVALID is deasserted, it indicates that MIILF and PRSD[15:0] are valid. The meaning of NVALID is valid only during a SCAN operation.

PRSD[15:0] **PHY Read Status Data** **Output**

The PRSD[15:0] signals contain the Status register data that is read from the addressed MII PHY. The

PRSD[15:0] signals are valid after the MIIM core deasserts BUSY for a read operation and after the MIIM core deasserts NVALID for a scan operation. See Section 3.2.3.2, “Read PHY Status Register,” on page 3-21 for a definition of the PHY Status register.

RGAD[4:0] **Register Address [4:0]** **Input**
The RGAD[4:0] signals contain the address of the register within the PHY the host has selected for reading or writing. The host must not change the RGAD[4:0] signals while the MIIM core is asserting the BUSY signal; otherwise, the MIIM core may read the RGAD[4:0] signals incorrectly.

RSTAT **Request MII Status** **Input**
The host asserts the RSTAT signal to request that the MIIM function initiate a status read operation from the addressed MII PHY. RSTAT must be at least one HCLK pulse wide and must be asserted and deasserted synchronously with the rising edge of HCLK. The host must ensure that FIAD[4:0] and RGAD[4:0] are valid before it asserts RSTAT. The host must keep the RSTAT signal deasserted if the SCAN signal is asserted.

SCAN **Status Read Scan** **Input**
When the host asserts the SCAN signal, the MIIM core causes continuous status read operations from the addressed MII PHY. The host must ensure that FIAD[4:0], and RGAD[4:0] are valid before it asserts SCAN and must not change the signals during the entire scan operation. When the host deasserts SCAN, status read operations stop.

SCAN must be at least one HCLK pulse wide and must be asserted and deasserted synchronously with the rising edge of HCLK.

2.3.2 MIIM Core to PHY Signals

The signals that connect the MIIM core to the PHY are listed below. All signals are active HIGH unless otherwise indicated. Signal direction is with respect to the MIIM core. The standard IEEE MII signal name references are shown in brackets.

MDC **MIIM Data Clock** **Output**

The MDC [MDC] signal is sent by the MIIM block to the PHY as a timing reference for transfer of information on the MDI and MDO signal lines. MDC is an aperiodic signal (HIGH-to-LOW and LOW-to-HIGH transitions do not necessarily occur at regular intervals) that has no maximum HIGH or LOW times. The minimum HIGH and LOW times are 200 ns each.

MDI **MIIM Data In** **Input**

The PHY transfers status on the MDI signal to the MIIM core. The PHY places status information on MDI synchronously, and the MIIM block samples the information synchronously.

MDI is driven through an open collector or open drain circuit that enables either the MIIM block or the PHY to deassert the signal. The PHY must provide a resistive pull-up of 1.5 kΩ ±5% to maintain the signal in a HIGH value. The MIIM block must incorporate a weak pull-down of 2 kΩ ±5% on the MDI signal and thus may use the quiescent state of MDI to determine if a PHY is connected to the MII. If no PHY is present, the MDI signal will be LOW. If a PHY is present, the MDI signal will be HIGH.

The MDI and MDO signals are joined into one signal [MDIO] outside the MIIM core before being connected to the PHY. The MDOEN signal controls the direction of data transfer on the MDIO signal and whether the MDI signal or the MDO signal is active at a particular time.

MDO **MIIM Data Out** **Output**

The MIIM core transfers control information on the MDO signal to the PHY. The MIIM block places control information on MDO synchronously to HCLK, and the PHY samples the information synchronously. MDO is driven through an open collector or open drain circuit that allows either the PHY or the MIIM core to deassert the signal.

The PHY must provide a resistive pull-up of 1.5 kΩ ±5% to assert the signal.

The MDI and MDO signals are joined into one signal [MDIO] outside the MIIM core before being connected to the PHY. The MDOEN signal controls the direction of data transfer on the MDIO signal and whether the MDI signal or the MDO signal is active at a particular time.

MDOEN **MIIM Data 3-State Enable** **Output**

The MDOEN signal is a 3-state driver enable that controls the open-drain MDO output data signal. The MIIM core asserts MDOEN synchronously to HCLK whenever the MDO signal contains valid data. The MIIM core deasserts MDOEN to indicate that data is not valid.

The MDI and MDO signals are joined into one signal [MDIO] outside the MIIM core before being connected to the PHY. The MDOEN signal controls the direction of data transfer on the MDIO signal and whether the MDI signal or the MDO signal is active at a particular time.

Chapter 3
Core Descriptions

This chapter describes the E-110 core functions and consists of the following sections:

- Section 3.1, “Clock Operation”
- Section 3.2, “E-110 Core Operation”

Standard IEEE MII signal name references are shown in brackets.

3.1 Clock Operation

This section describes all of the clocks used in the E-110 core and their operation. The clocks fall into the following categories:

- Clocks from the PHY to the MAC core—MTXC and MRXC
- Clock from the host to the MIIM core—HCLK
- Clock from the MIIM core to the PHY—MDC

3.1.1 MTXC and MRXC

MTXC is the transmit nibble or symbol clock input from a PHY to the MAC. The MTXC signal operates at a frequency of 25 MHz or 2.5 MHz. MTXC is a continuous clock that provides a timing reference for transfer of the MTXEN, MTXD[3:0], and MTXER signals from the E-110 MAC to the PHY. The PHY generates MTXC.

The MTXC frequency is 25% of the transmit data rate. A PHY operating at 100 Mbit/s provides an MTXC frequency of 25 MHz ± 100 ppm. A PHY operating at 10 Mbit/s provides a TX_CLK frequency of 2.5 MHz ± 100 ppm. The duty cycle of the TX_CLK signal is between 35% and 60%.

The MRXC [RX_CLK] signal is a continuous clock that provides a timing reference for transfer of the MRXDV, MRXD[3:0], and MRXER signals from the PHY to the E-110 MAC. MRXC is an input from the PHY.

As long as the PHY is receiving a continuous signal from the medium and can recover the MRXC clock reference and supply MRXC, the PHY does not need to make a switch from the recovered clock reference to a nominal clock reference (for example, the MTXC continuous clock signal from the PHY). However, if the loss of a receive signal causes the PHY to lose the recovered clock reference, the PHY must source MXRC from a nominal clock reference.

If the PHY needs to make a switch from recovered clock to nominal clock, it deasserts the MRXDV signal. During the interval between MCRS and the assertion of MRXDV at the beginning of a frame, the PHY may extend a cycle of the MRXC clock by holding it in either the HIGH or LOW condition until the PHY has successfully locked onto the recovered clock. Successive cycles of MRXC must meet the duty cycle requirement.

While MRXDV is asserted, the PHY recovers MRXC from the received data. MRXC has a frequency equal to 25% of the data rate of the received signal and is synchronous to recovered data. The duty cycle is between 35% and 60% and is nominally 50%. For 100-Mbit/s operation, MRXC is nominally 25 MHz ±100 ppm, and the minimum MRXC HIGH and LOW times are 14 nsec. For 10-Mbit/s operation, MRXC is nominally 2.5 MHz ± 100 ppm, and the minimum MRXC HIGH and LOW times are 140 ns.

When the MCRS signal is deasserted, the PHY provides MRXC at the PHY’s nominal clock rate (for example, the MTXC clock signal) and with nominal duty cycle. The minimum HIGH and LOW times are each 35% of the nominal MRXC period except for the transition between recovered clock frequency and nominal clock frequency, which occurs while MRXDV is deasserted. Following the transition from MRXDV asserted to MRXDV deasserted, the PHY can keep MRXC in either the HIGH or LOW condition to extend the MRXC clock by one cycle until the PHY is ready to provide MRXC from a nominal clock source. The maximum HIGH or LOW time for MRXC during this transition is two times the nominal clock period (a total of 80 ns for 25-MHz operation or 800 ns for 2.5-MHz operation).

3.1.2 HCLK

The host clock is either a 33-MHz or 25-MHz clock. The CLKS signal indicates the HCLK frequency to the MIIM core. The host clock generates the MIIM Data Clock (MDC), which has minimum HIGH and LOW times of 200 ns each. A 33-MHz HCLK is divided by 14 to generate a 2.35 MHz MDC and a 25-MHz HCLK is divided by 10 to generate a 2.5 MHz clock. To meet the minimum HIGH and LOW times, the maximum frequency of MDC is 2.5 MHz.

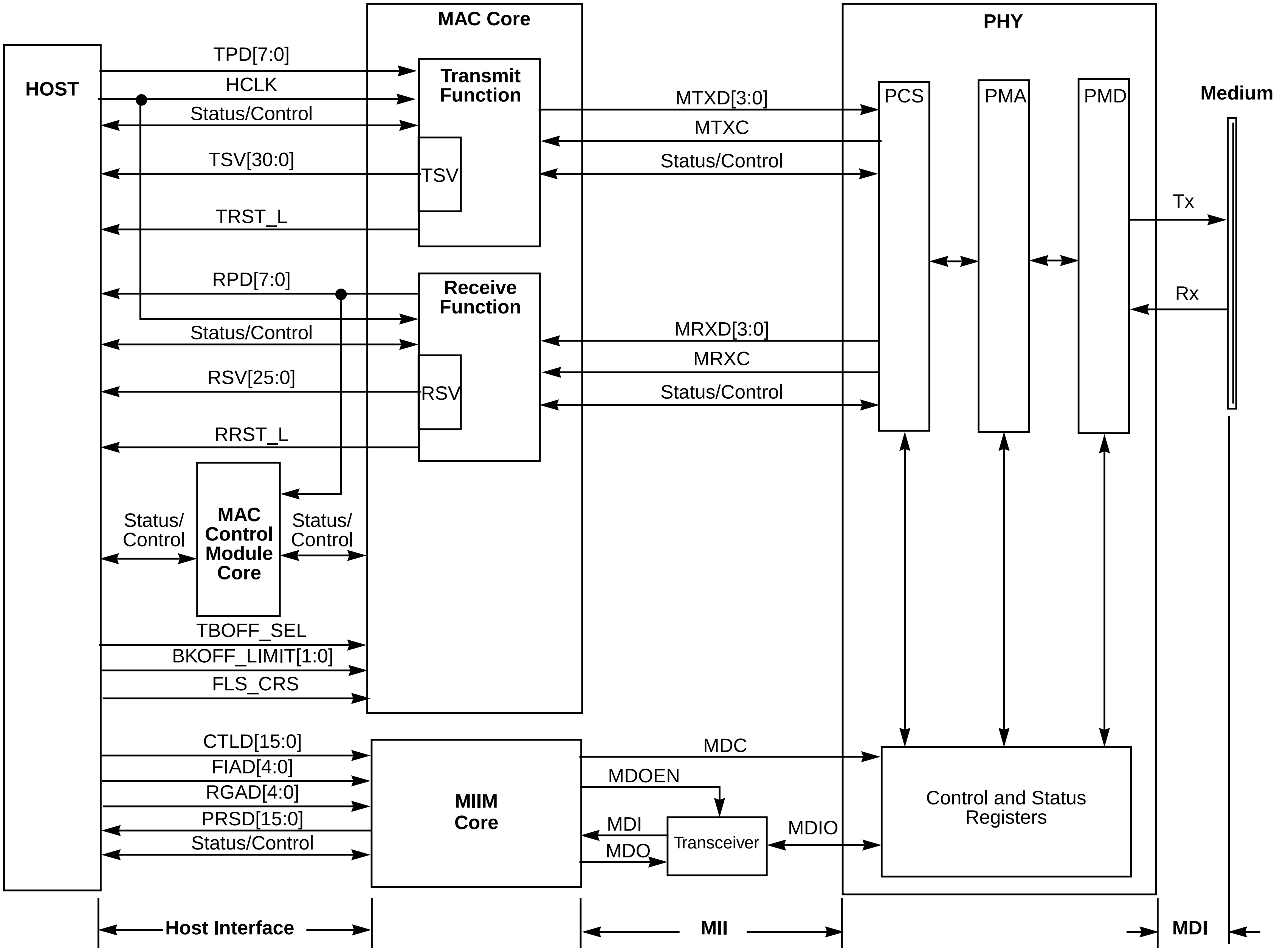
3.1.3 MDC

The MDC signal is the MIIM core data clock output. The MDC [MDC] signal is sent by the core to the PHY as a timing reference for transfer of information on the MDI and MDO signal lines. MDC is an aperiodic signal that has no maximum HIGH or LOW times. The minimum HIGH and LOW times are 200 ns each.

3.2 E-110 Core Operation

The following subsections describe the operation of the E-110 core. Figure 3.1 is an overall block diagram, showing how the MAC receive and transmit functions, MIIM core, and MAC control module core interface to the host and the PHY.

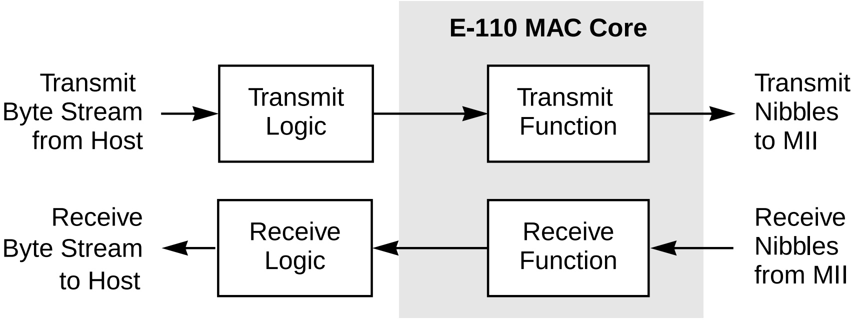
Figure 3.1 Overall Block Diagram



3.2.1 MAC Core Operation

Figure 3.2 shows the E-110 MAC core transmit and receive function blocks, along with customer-defined transmit and receive logic blocks.

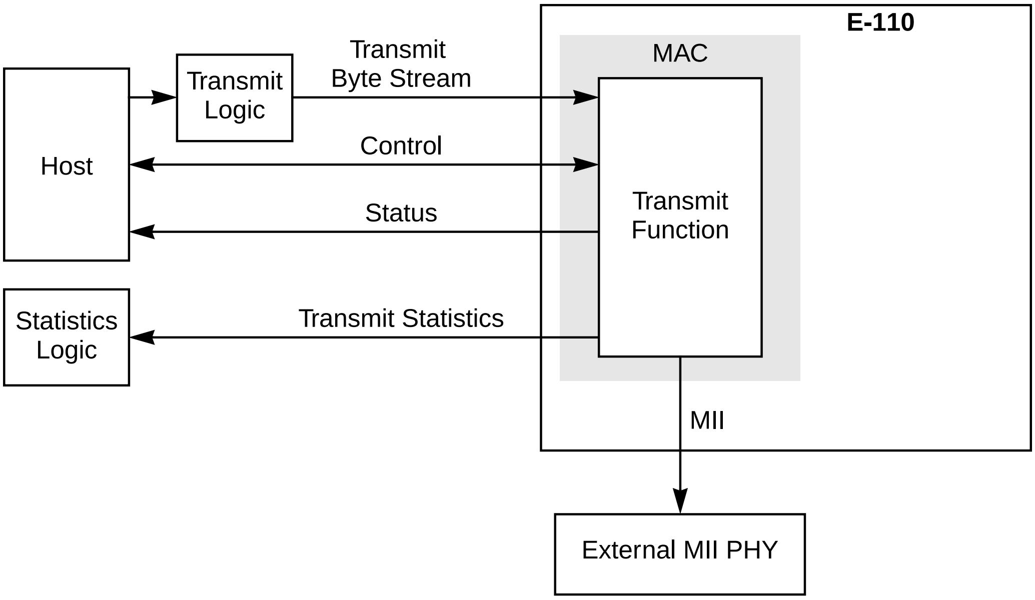
Figure 3.2 Transmit and Receive Functions



3.2.1.1 MAC Transmit Function

The normal E-110 MAC architecture includes an independent transmit function between the transmit logic on the host side and the MII, which is located on the PHY side. The transmit logic block is customer-defined and buffers data from the host. The E-110 MAC transmit function block performs the E-110 MAC operations. Figure 3.3 shows how the transmit function interfaces to the rest of the system.

Figure 3.3 Transmit Function Interface



The general flow of transmit packet data is from the host through the transmit control logic to the transmit function block and then into the media PHY device. The transmit function block of the E-110 MAC generates 100BASE-T transmit MII nibble data streams in response to byte streams supplied from the transmit logic. The transmit function performs the required deferral and backoff algorithms and reports per packet statistics. The transmit function operates on the 25- or 2.5-MHz MII transmit nibble clock.

The E-110 MAC core monitors the physical medium even when it has nothing to transmit by tracking the Carrier Sense (CRS) signal from the PHY. When the medium is busy, the MAC defers to the frame in progress by delaying any pending transmission of its own. When the PHY deasserts CRS, it indicates to the MAC that the last bit of the frame has been transmitted onto the physical medium. The MAC then continues to defer transmission for a period called the interpacket gap (IPG).

The purpose of *deference* is to ensure a minimum interframe spacing to allow a station recovery time to process a received frame and for the medium to recover. Deference applies to both 10-Mbit/s and 100-Mbit/s operation.

There is an IPG counter inside the MAC that begins counting after a deferral. You can specify three threshold values for the counter:

- IPGR1[6:0]—Non Back-to-Back IPG, Part 1
- IPGR2[6:0]—Non Back-to-Back IPG, Part 2
- IPGT[6:0]—Back-to-Back Transmit IPG

To provide maximum flexibility for the IPG in both full-duplex and half-duplex operation, these thresholds are programmable. Due to possible variations in PHY propagation delays, the programmable thresholds may need to be adjusted to meet IEEE 802.3u deference requirements.

As soon as the CRS signal is deasserted, the MAC begins timing the IPG with the IPG counter.

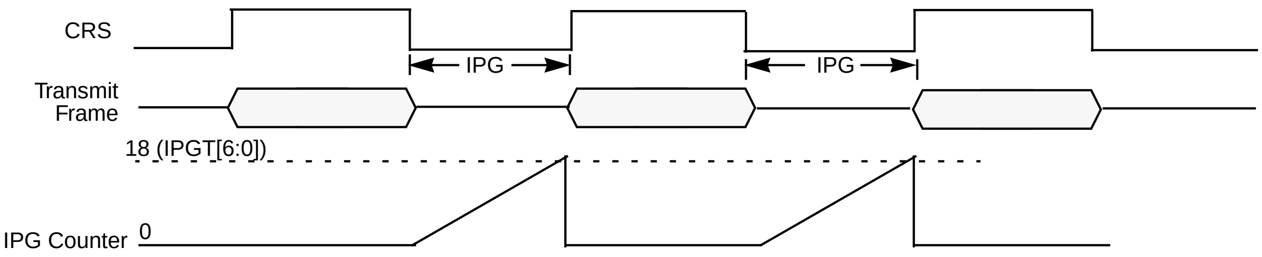
IPG Half-Duplex Operation – The IPGT[6:0][1] signals contain the threshold value to be compared to the IPG counter. The threshold value is used for back-to-back MAC transmissions. As soon as the MAC’s IPG count is equal to IPGT[6:0], the MAC may begin transmission.

IPGR1[6:0] is the threshold value used to reset the MAC's IPG counter if CRS is detected within a short time after receiving a packet (see Figure 3.5). If CRS is asserted before the IPG counter value reaches the IPGR1[6:0] threshold, the MAC resets the IPG counter because the CRS may have occurred due to a collision fragment (see Figure 3.5). If the IPG counter makes it past the IPGR1[6:0][1] threshold, the MAC does not reset the IPG counter so that the E-110 MAC is able to access to the medium without being constantly deferred by other stations trying to gain access.

IPGR2[6:0][2] is the threshold value for measuring the IPG if the MAC has been receiving and the host wants the MAC to transmit. That is, when the IPG counter has exceeded the value in IPGR1[6:0] and reaches the value in IPGR2[6:0], the MAC can transmit.

Because IPGR1[6:0] and IPGR2[6:0] are programmable, IPGR1[6:0] may be set to a threshold that is a fraction of IPGR2[6:0] to conform to the IPG rules of the IEEE 802.3/802.3u specifications.[3] Figure 3.4 and Figure 3.5 illustrate half-duplex IPG operation.

Figure 3.4 Half-Duplex Back-to-Back IPG Operation



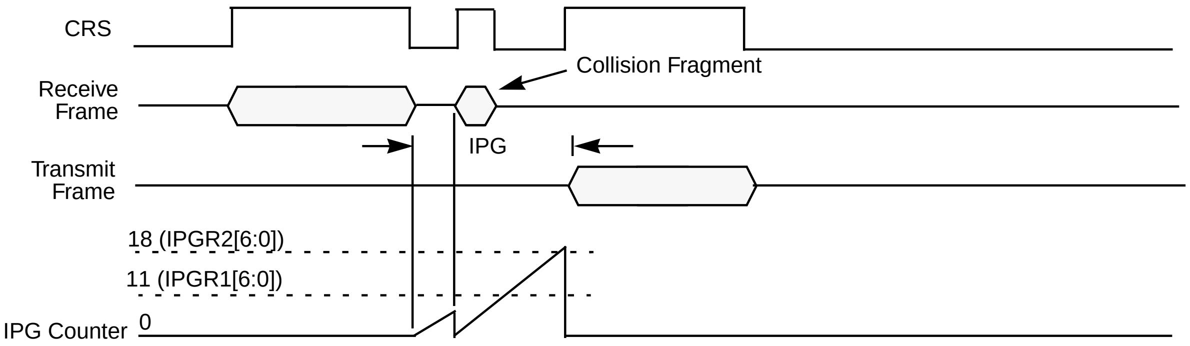
[1] A value of 18 (decimal) for IPGT[6:0] results in an IPG of 960 ns for 100-Mbit/s operation (the value given in IEEE 802.3u, section 1.4.102) at the MII. When IPGT[6:0] is set to 18, the IPG counter counts from 0 to 18 (19 clocks total) using the nibble clock. The nibble clock runs at 25 MHz (40 ns) for 100-Mbit/s operation and 2.5 MHz (400 ns) for 10-Mbit/s operation). The 19 clocks plus a 5-clock internal core delay results in 24 clocks, which yields 960 ns for 100-Mbit/s operation or 9.6 µs for 10-Mbit/s operation. Depending on PHY propagation delays, the value for IPGT[6:0] may need to be adjusted to meet the IPG requirement at the network interface.

[1] The nominal value for IPGR1[6:0] is 11 (decimal).

[2] A nominal value for IPGR2[6:0] is 18 (decimal). The PHY may have its own propagation delay and the value of 18 (decimal) may or may not be the correct value for a particular PHY.

[3] See IEEE 802.3 Sections 4.2.3.2.1 and 4.2.3.2.2. See also IEEE 802.3u Section1.4.102.

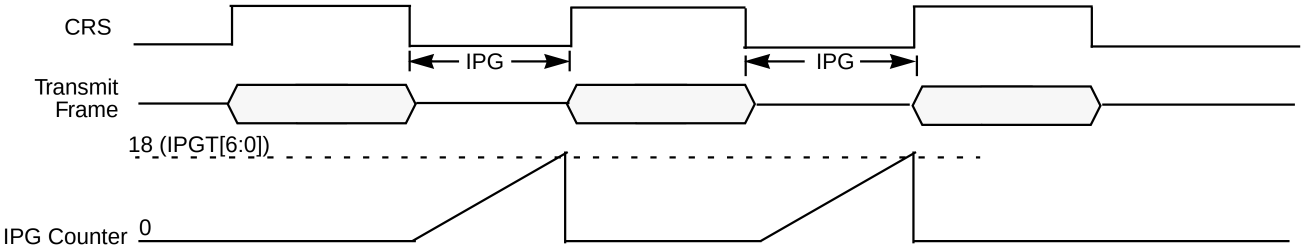
Figure 3.5   Half-Duplex Non Back-to-Back IPG Operation



IPG Full-Duplex Operation – In full-duplex mode, the MAC never sees the CRS signal asserted because CRS is disabled when the FULLD signal is asserted. For this reason, in full-duplex mode the MAC does not defer on CRS. In full-duplex mode, the MAC can begin to create the IPG as soon as transmission is done, regardless of the state of CRS.

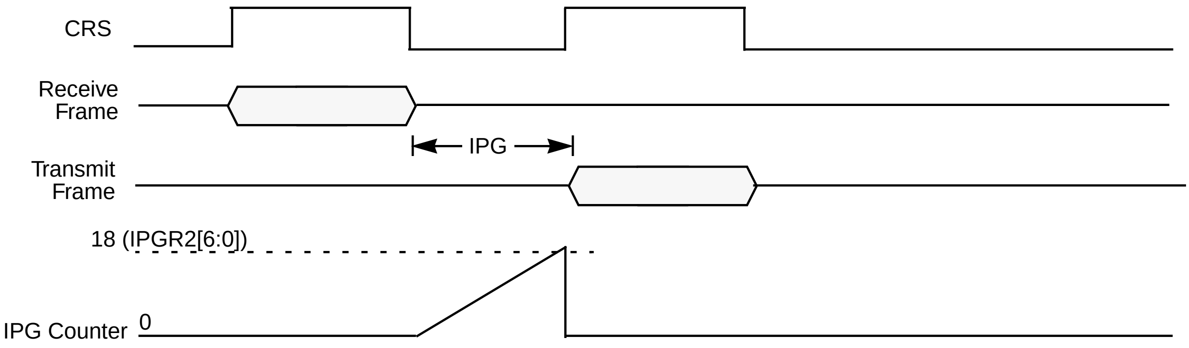
When the MAC has been receiving and the host wants the MAC to transmit a packet, the MAC looks at the IPGR2[6:0] threshold. The MAC compares the IPG counter to the IPGR2[6:0] threshold in situations where a reception is followed by a transmission. When the host wants to perform back-to-back transmit operations through the MAC, the MAC compares the IPG counter to the IPGT[6:0] threshold. For full-duplex operation, the IPGR2[6:0] threshold should normally be set equal to IPGT[6:0].[1] Figure 3.6 and Figure 3.7 illustrate full-duplex IPG operation.

Figure 3.6   Full-Duplex Back-to-Back IPG Operation



[1] A value of 18 (decimal) for IPGR2[6:0] and IPGT[6:0] results in an IPG of 960 ns at the MII for 100-Mbit/s operation and 960 µs for 10-Mbit/s operation.

Figure 3.7 Full-Duplex Non Back-to-Back IPG Operation



When instructed, the transmit function adds a preamble and SFD to the supplied packet data and, if requested, appends a generated FCS. The E-110 MAC transmit function adds bytes of zero before the appended FCS to pad packets with fewer than 60 data bytes. The transmit function detects and enforces collisions with jams and manages transmission retries.

The transmit function block operates on the 25-MHz (100-Mbit/s operation) or 2.5-MHz (10-Mbit/s operation) transmit nibble or symbol clock (MTXC) supplied by the PHY. The PHY clock must be a continuous clock. The connecting interface to the transmit control logic also operates on the same transmit nibble clock and communicates synchronously with the transmit function.

The transmit data is supplied to the transmit function a byte at a time from the transmit control logic on the TPD[7:0] signals and output to the MII a nibble at a time as defined in the MII specification. Transmit data bytes are transferred from the host at one-half the nibble clock rate. The transmit data is also sent to the transmit function block FCS generator/checker. The MII output nibbles (MTXD[3:0]) are only valid when the MTXEN signal is asserted. The transmit function supplies nibbles of zero when MTXEN is not asserted. When MTXEN is asserted, the transmit function supplies preamble, SFD, transmit data, FCS, jam, or pad nibbles.

The operation of the E-110 MAC transmit function is affected by the following control signals: no preamble (NOPRE), pad enable (PADEN), full-duplex (FULLD), FCS enable (CRCEN), and huge packet enable (HUGEN). The host must appropriately manage any transitions of these signals and ensure that only valid combinations of controls signals are used, as described below.

- NOPRE. If no preamble is selected (the NOPRE signal is asserted), the preamble and start of frame nibbles come directly from the host and are not examined by the E-110 MAC transmit function. If NOPRE is not asserted, the MAC encodes the preamble including the SFD and outputs it at the beginning of the packet.
- PADEN. When asserted, the PADEN signal instructs the transmit function to pad packets of fewer than 60 bytes with bytes of zero. The CRCEN signal must also be asserted because the MAC must calculate the FCS when it inserts its own pad bytes. If PADEN is not asserted, the host must ensure that packets contain at least 60 bytes; otherwise, the packet is invalid (too short).
- FULLD. When asserted, the FULLD signal instructs the transmit function to operate in full-duplex mode by not deferring to the CRS signal and not responding to the COL signal.
- CRCEN. If the CRCEN signal is deasserted, the last four data bytes of a packet must be a valid FCS provided by the host and are checked by the MAC's transmit function. If the FCS is not valid, the MAC asserts the FCS error signal (TSV21) in the Transmit Statistics Vector. If CRCEN is asserted, the transmit function generates an FCS from the transmit data bytes and pad bytes of the packet and outputs it at the end of the packet. You can change CRCEN on a packet-by-packet basis, but you must not change it during an active transmission.
- HUGEN. When asserted, the HUGEN signal instructs the MAC transmit function to allow transmission of packets up to 32 Kbytes rather than the normal limit of 1518. However, the MAC core does not enforce the length of the maximum packet. Instead, the higher layer protocols must assure that the packet size does not exceed IEEE specifications.

For more information about these signals, refer to Chapter 2.

A packet transmission from the host to the transmit function begins when the host asserts the TPSF signal. When the TPSF signal is asserted, the host may also present the first data byte of the packet on TPD[7:0]. The host may also delay supplying the first data byte on TPD[7:0] until the transmit function has sent out the preamble.

Assuming the NOPRE signal is not asserted, the transmit function (after any pending backoff and needed deference) begins to generate preamble and start of frame nibbles. After generating the SFD, the transmit function uses the contents of TPD[7:0] as the first data byte and asserts the TPUD signal to the host. If the host had not already supplied the first data byte, it must recognize an underrun condition and assert TPUR, which causes the transmit function to abort the packet. The host thus has at least eight byte times (the length of the preamble plus SFD) to supply the first data byte after asserting TPSF.

Conversely, the TPUD signal might not be asserted for over 500,000 bit times in the case of a maximum backoff and deferral. After once seeing TPUD asserted, the host must then deassert TPSF and supply new transmit packet data bytes every other MTXC clock cycle. The host continues supplying new data bytes up until the last data byte of the packet, when it asserts the TPEF signal. If the host is unable to correctly supply new transmit packet data bytes, the transmit function must assert TPUR. When the host asserts the TPUR signal, the MAC asserts the MTXER signal with the last nibble it transmits to indicate that this nibble may not have valid information. Each data byte must be stable on the TPD[7:0] signals for two transmit clock periods. The TPEF signal must also be valid for two transmit clock periods. The TPSF signal and the first data byte must be kept asserted for one transmit clock period after the TPUD signal is asserted.

Alternatively, if NOPRE is asserted, the host must supply a data byte as soon as it asserts TPSF. The transmit function does not supply a preamble when NOPRE is asserted, so the preamble must come from the host. The host must supply the first valid data byte on the TPD[7:0] signals as soon as it asserts TPSF.

The packet is terminated when the host asserts the TPEF signal at the end of the frame, which causes the transmit function to pad the packet with bytes of zero if necessary (provided the PADEN signal is asserted) and append the FCS (provided the CREN signal is asserted), then terminate transmission. The transmit function asserts the TPDN signal when the transmission is successfully completed. Packets longer than 1518 bytes (HUGEN signal deasserted) or 32 Kbytes (HUGEN signal asserted) are truncated. Any collision causes the transmission to be truncated and extended with jam bytes of zeroes to ensure that the entire network sees the collision.

Collision detections other than a 16th collision without an intervening non-colliding transmission cause the TPRT signal to be asserted, which informs the host to restart the packet. Late collisions are treated just the same as any other collisions if the RTRYL control input is asserted; otherwise, a late collision causes an abort.

TPAB is asserted when any of the following conditions exists:

- A 16th collision occurs without an intervening non-colliding transmission.
- A packet transmission attempt is made that has excessive deferrals (greater than 6,072 transmit nibble clocks when the HUGEN signal is deasserted or greater than 131,072 transmit nibble clocks when HUGEN is asserted).
- A transmission occurs with *underflow*. (The host did not supply new transmit packet bytes fast enough to keep up with the MAC transmit function.)

When the TPAB signal is asserted, the host must discard the packet and try again with the next packet available.

A subsequent packet transmission, indicated by the assertion of the TPSF signal, deasserts the TPDN), TPRT, and TPAB signals.

Transmit Function to Host Interface – The interface between the E-110 MAC transmit function and the host consists of the following:

- Transmit control logic
- Transmit byte stream signals
- Control signals between the MAC and host

The transmit control logic serves two basic purposes. It contains temporary storage for transmit bytes from the host in the form of buffers or FIFOs, and it translates host signals to transmit byte stream signals that are compatible with the MAC transmit function signals. For details on the MAC transmit function interface, see Section 2.1.1, “Transmit Function to Host Signals,” on page 2-3.

The transmit byte stream from the host to the transmit function consists of an eight-bit parallel data interface consisting of the TPD[7:0] signals.

The control signals from the host to the MAC’s transmit function are TPSF, TPEF, and TPUR.

The control signals from the MAC's transmit function to the host are TPUD, TPDN, TPRT, TPAB, and TRST_L.

Transmit Function to Statistics Interface – The interface between the transmit function and external statistics collection logic consists of the TSV[30:0] and TSVP_L signals. For more details on these signals, see Section 2.1.6, "Statistics Vector to Host Signals," on page 2-21.

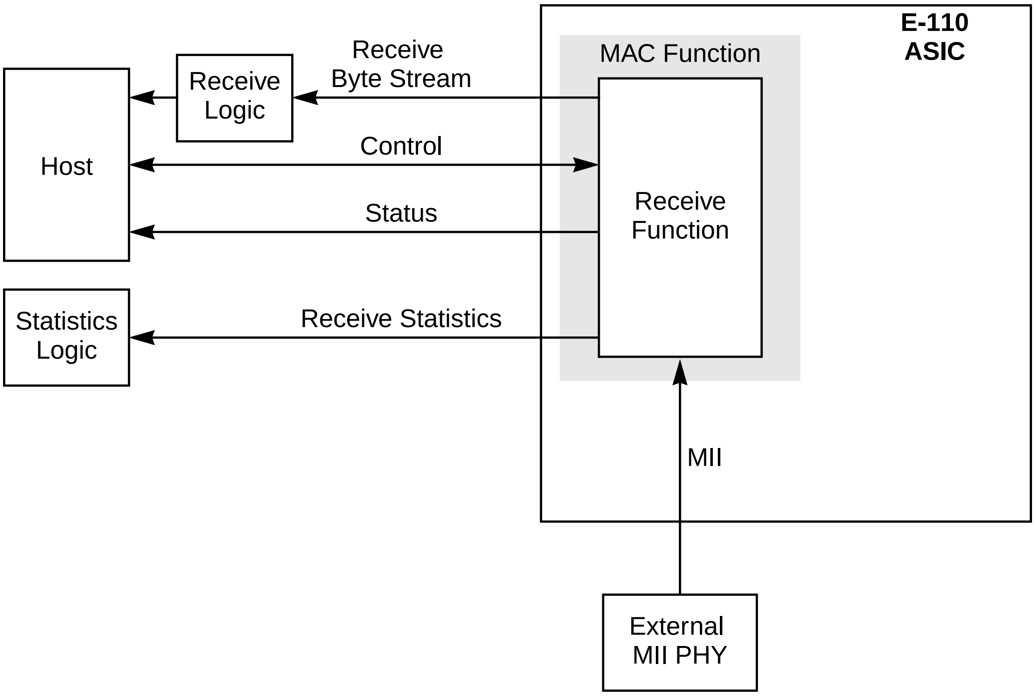
Transmit Function to MII Interface – The MII interface of the transmit function allows the E-110 MAC to be compatible with MII-supported PHYs. The MII interface signals conform to the IEEE 802.3u standard for 100-Mbit/s baseband networks.

The MII interface consists of a set of transmit signals and a set of receive signals. The transmit signals are MTXD[3:0], MTXEN, MTXER, MCRS, MCOL, and MTXC. For more details on these signals, see Section 2.1.4, "MII to PHY Signals," on page 2-10.

3.2.1.2 MAC Receive Function

The normal E-110 MAC architecture includes an independent receive channel between the host and the MII, which is located on the PHY side. The receive function block provides the Ethernet MAC operations described in this section. Figure 3.8 shows how the receive function interfaces to the rest of the system.

Figure 3.8 Receive Function Interface



The general flow of receive packet data is a from a PHY into the receive function block and out through the receive logic to the host system. The receive function block of the E-110 MAC interprets 100BASE-T MII receive data nibble streams and supplies correctly formed packet byte streams to the host.[1] The receive function searches for the SFD at the beginning of the packet, verifies the FCS, and detects any dribble nibbles or receive code violations.

After any packet has been sent to the host from the receive function, the MAC produces a receive statistics vector. A long event or a carrier event is noted for a subsequent statistics vector. See the definition of the RSV[25:0] signals in Chapter 2 for an explanation of long and carrier events.

The receive function detects and removes the preamble and SFD bytes at the beginning of the received packet. The E-110 MAC receive function enforces a minimum IPG between the end of the data portion of one packet, which contains the FCS, and the beginning of the data (destination address) portion of the following packet. The IPG allows for the stations and media to recover between transmissions.

[1] See Section 1.2.1, “Fast Ethernet Overview,” on page 1-2 for an explanation of 100BASE-T.

The receive function block operates on the 25-MHz (100-Mbit/s operation) or 2.5-MHz (10-Mbit/s operation) transmit nibble or symbol clock (MRXC) supplied by the PHY. MRXC is considered to be a continuous clock although it may have gaps as defined in the IEEE 802.3u MII specification. The connecting interface to the host also operates on the same receive nibble clock and communicates synchronously with the receive function. Refer to Section 2.1.2, “Receive Function to Host Signals,” on page 2-5 for detailed requirements on how the host interfaces to the receive function.

The receive function supplies the receive data a byte at a time to the host on the RPD[7:0] signals and inputs the receive data from the PHY a nibble at a time. Receive data bytes are transferred at one-half the nibble clock rate. The receive nibble data is also sent to the receive function FCS checker. MII input nibbles (MRXD[3:0]) are only valid when MRXDV is asserted.

The FULLD and HUGEN control signals affect the operation of the E-110 MAC receive block. The host must appropriately manage any transitions of these signals and ensure that only valid combinations of controls signals are used. The E-110 MAC receive function also responds to the MTXEN signal from the transmit function when not in full-duplex mode. If MTXEN is asserted, it indicates that a data packet is being transmitted. If the MAC is in half-duplex mode, the receive function is disabled when MTXEN is asserted and enabled when it is deasserted. In full-duplex mode, the MTXEN signal does not affect the receive function.

A packet transfer to the host from the receive function begins when the receive function asserts the RPSF and RPDV signals along with the first byte of received packet data after removing the preamble and SFD. For subsequent data bytes, the receive function asserts only the RPDV signal until the last byte, when it asserts both RPDV and RPEF. There is no provision for flow control feedback from the host to the receive function, so the host must handle the receive packet data bytes as they arrive.

The receive packet starts at the MII interface when the PHY asserts the MRXDV signal and ends when the PHY deasserts the MRXDV signal. The receive function passes packets less than or equal to 1518 bytes (or less than or equal to 32 Kbytes with the HUGEN signal asserted). Longer packets are truncated. Collision fragments and other carrier events (for example, a preamble but no SFD) are received as packets if a valid preamble is seen following a valid IPG. The MAC asserts the RSV25 signal, one of the bits of the Receive Statistics Vector, to indicate that a carrier event occurred. The receive function notes a long event if the MCRS signal is asserted for over 24,288 bit times if the HUGEN signal is not asserted and over 524,288 bit times if HUGEN is asserted.

Other modules can use the RRST_L signal as a host reset synchronized to the MRXC receive clock. Because the MRXC signal can be slow with respect to a host reset pulse, or even stopped,[1] the MAC latches the host reset signal, HRST_L, without using MRXC and uses the HRST_L signal to assert RRST_L asynchronously to MRXC. The MAC then deasserts RRST_L synchronously on the transition of the MRXC clock.

Receive Function to Host Interface – The interface between the E-110 MAC receive function and the host consists of the following:

- Receive control logic
- Receive byte stream signals
- Status Signals

The receive control logic is a designer-specific implementation. It may contain temporary storage in the form of buffers or FIFOs for receive bytes from the MAC, and it may translate receive byte stream signals from the receive function that are compatible with the host interface signals.

The host receives the byte stream from the receive function over an eight-bit parallel data interface consisting of the RPD[7:0] signals.

The status signals from the receive function to the host are RPDV, RPSF, RPEF, RRST_L, CRCO[9:1], CRCG, BCO, MCO, and BYTE7. For more details on these signals, see the subsection entitled “Receive Function to Host Signals” on page 2-5.

Receive Function to Statistics Interface – The interface between the receive function and external statistics collection logic consists of the RSV[25:0] and RSVP_L signals. For more details on these signals, see Section 2.1.6, “Statistics Vector to Host Signals,” on page 2-21.

[1] MRXC from the PHY may be extended by a cycle when the PHY switches from recovered clock to nominal clock. See the subsection entitled “MRXC Receive Nibble or Symbol Clock Input” on page 2-11.

Receive Function to MII Interface – The MII of the receive function allows the E-110 MAC to be compatible with MII PHYs. The MII interface signals conform to the IEEE 802.3u standard for 100 Mbit/s baseband networks.

The MII consists of a set of transmit signals and a set of receive signals. The receive signals are MRXD[3:0], MRXDV, MRXER, and MRXC. For more details on these signals, see Section 2.1.4, “MII to PHY Signals,” on page 2-10.

3.2.2 MAC Control Module Core Operation

The optional MAC control module core implements full-duplex flow control for the E-110 core.

Control frames currently have one defined opcode: pause. The pause operation is used to inhibit transmission of data frames for a specified period of time. Using the MAC control module core, a client wishing to inhibit transmission of data frames from another station generates a pause control frame. The pause control frame contains a reserved multicast address of 01-80-C2-00-00-01, a pause opcode, and a pause time field, consisting of a 16-bit value expressed in slot times. The pause operation does not inhibit the transmission of MAC control frames but it does inhibit transmission of MAC data frames until the pause timer expires. Pause frames, however, do not affect the E-110 core data transmission until frames currently being transmitted are finished.

Upon receipt of a valid MAC control frame with the pause opcode and a destination address equal to a multicast address of 01-80-c2-00-00-01 or its own unicast address, the MAC control module loads its pause timer with the value sent in pause time field. If the pause Time field is non-zero, the E-110 core is said to be paused from transmitting data frames and waits for pause time number of slot times to initiate transmission. New pause frames override previous pause frames. The MAC control module core does not update its pause timer value when it receives an invalid control frame (one containing a bad FCS, code errors, or dribble nibbles or a control packet less than minimum packet size).

The MAC control module reports statistics for management purposes. Along with the RSV[25:0] and TSV[30:0] statistics vector signals, the XMT_PAUSEF, RSVD_PAUSEF and UNSUPP_OPCODE signals are available to the host to read for management purposes.

A mirror counter in the MAC control module core is loaded with a value when a pause control packet is sent. The purpose of the mirror counter is to roughly keep track of the other station's pause timer. The mirror value has to take into account the packet transmit time and the pause value load time on the other end. The packet transmit time is the time it takes for the station's own control packet to be transmitted. The pause value load time is the time it takes for the other station to decode the packet, check the FCS, and load the pause time into its pause timer. The station receiving the control packet loads X (pause time) into the pause timer and the station transmitting the control packet loads X+ n (n is a programmable value) into the mirror counter. The mirror counter allows the transmitting station to send another pause control packet before the receiving station's pause timer expires if the receive buffer congestion has not been alleviated.

3.2.3 MIIM Core Operation

The optional MIIM core is a 16-bit parallel interface on the host side and a 4-signal interface (MDO, MDI, MDOEN, and MDC) on the MII side. The MIIM controls a PHY and gathers status from it. The register layout for a PHY is shown in Table 3.1. All PHYs that incorporate an MII contain the basic register set (registers 0 and 1). Registers 2 through 7 are part of the extended register set.

Table 3.1 MIIM Register Set

Register Address	Register Name	Basic/Extended
0	Control	Basic
1	Status	Basic
2–3	PHY Identifier	Extended
4	Auto-Negotiation Advertisement	Extended
5	Auto-Negotiation Link Partner Ability	Extended
6	Auto-Negotiation Expansion	Extended
7	Auto-Negotiation Next Page Transmit	Extended
8–15	Reserved	Extended
16–31	Vendor-Specific	Extended

3.2.3.1 Write PHY Control Register

To write the Control Register in the PHY, the host places the PHY address on the MIIM core's FIAD[4:0] signals and the address of the Control Register within the PHY on the core's RGAD[4:0] signals. The host keeps the RSTAT signal deasserted during a write operation.

The host CTLD[15:0] signals contain the data that is sent to the PHY Control Register. When the host asserts the LCTLD signal, the data on the FIAD[4:0], RGAD[4:0], and CTLD[15:0] buses are loaded into the MIIM core, and the core asserts the BUSY signal. The host must not change any of the address or data lines while BUSY is asserted. During the time the MIIM core asserts BUSY, it uses the MDO, MDC, and MDOEN signals to transport data to the Control Register of the selected PHY, using the *Management Frame Format* specified in the MII specification. The MDO and MDI signals are normally connected to a 3-state bidirectional *transceiver* on the PHY side of the MIIM core. A single bidirectional data line from the transceiver, MDIO, transfers control and status data to and from the PHY. The MDOEN signal controls the transceiver direction.

The PHY Control Register (Register 0) layout is shown in Table 3.2.

Table 3.2 PHY Control Register Bit Definitions

Bit	Name	Description	Operation
15	Reset	1 = PHY reset 0 = Normal operation	R/W,[1] SC[2]
14	Loopback	1 = Enable loopback[3] 0 = Disable loopback	R/W
13	Speed Selection	1 = 100 Mbit/s 0 = 10 Mbit/s	R/W
12	Auto-Negotiation Enable	1 = Enable auto-negotiation[4] 0 = Disable auto-negotiation	R/W
11	Power Down	1 = Power down 0 = Normal operation	R/W
10	Isolate	1 = Electrically isolate PHY from MII 0 = Normal operation	R/W
9	Restart Auto-Negotiation	1 = Restart auto-negotiation process 0 = Normal operation	R/W, SC
8	Duplex Mode	1 = Full-duplex 0 = Half-duplex	R/W
7	Collision Test	1 = Enable COL signal test 0 = Disable COL signal test	R/W
6–0	Reserved	Write as 0, ignore on read	R/W

1. R/W = Read/Write.
2. SC = Self-Clearing. (After the bit is set to one, the operation takes place and the PHY clears the bit back to zero.)
3. Loopback connects the PHY transmit signals to the PHY receive signals, which allows network isolation and troubleshooting.
4. Auto-negotiation is a signalling method described in clause 28 of the IEEE 802.3u specification. It allows each node to define its own operating mode (for example, 10 Mbit/s, 100 Mbit/s, 100BASE-TX, 100BASE-T4, or full-duplex) and to determine the operating mode of the node to which it is connecting. Auto-negotiation is conceptually similar to the signalling method used with modems, where two nodes negotiate their capabilities and settle on the highest common denominator.

3.2.3.2 Read PHY Status Register

To read the Status Register in the PHY, the host places the PHY address on the MIIM core's FIAD[4:0] signals and the address of the desired PHY register on the RGAD[4:0] signals. The host asserts RSTAT to indicate a read operation.

When the host asserts the RSTAT signal, the MIIM core loads the FIAD[4:0] and RGAD[4:0] address information into the core and the core asserts the BUSY signal. The host must not change any of the address lines while BUSY is asserted. While the MIIM core asserts BUSY, it uses the MDO, MDC, and MDOEN signals to read the PHY Status Register.

The PHY responds to the read request and sends data to the MIIM core by means of the MDI signal. After the MIIM core has received all of the data from the PHY Status Register, it deasserts the BUSY signal and places the data on the PRSD[15:0] signals to the host. Data is transferred to and from the PHY using the Management Frame Format specified in the MII specification.

The PHY Status Register (Register 1) layout is shown in Table 3.3.

Table 3.3 PHY Status Register Bit Definitions

Bit	Name	Description	Operation
15	100BASE-T4[1]	1 = PHY able to perform 100BASE-T4 0 = PHY not able to perform 100BASE-T4	R/O[2]
14	100BASE-TX[3] Full-Duplex	1 = PHY able to perform full-duplex 100BASE-TX 0 = PHY not able to perform full-duplex 100BASE-TX	R/O
13	100BASE-TX Half-Duplex	1 = PHY able to perform half-duplex 100BASE-TX 0 = PHY not able to perform half-duplex 100BASE-TX	R/O
12	10 Mbit/s Full-Duplex	1 = PHY able to operate at 10 Mbit/s in full-duplex mode 0 = PHY not able to operate at 10 Mbit/s in full-duplex mode	R/O
11	10 Mbit/s Half-Duplex	1 = PHY able to operate at 10 Mbit/s in half-duplex mode 0 = PHY not able to operate at 10 Mbit/s in half-duplex mode	R/O
10–6	Reserved	Ignore when read	R/O
(Sheet 1 of 2)			

Table 3.3 PHY Status Register Bit Definitions (Cont.)

Bit	Name	Description	Operation
5	Auto-Negotiation Complete	1 = Auto-negotiation process completed 0 = Auto-negotiation process not completed	R/O
4	Remote Fault	1 = Remote fault condition detected 0 = No remote fault condition detected	R/O, LH[4]
3	Auto-Negotiation Ability	1 = PHY is able to perform auto-negotiation 0 = PHY is not able to perform auto-negotiation	R/O
2	Link Status	1 = Link is up 0 = Link is down	R/O, LL[5]
1	Jabber Detect	1 = Jabber condition detected. Jabber is defined as a condition on an Ethernet network when a node transmits for longer than the specified time. 0 = No jabber condition detected	R/O, LH
0	Extended Capability	1 = Extended register capabilities 0 = Basic register capabilities	R/O
(Sheet 2 of 2)			

1. 100BASE-T4 allows users to run 100BASE-T over four pairs of Category 3, 4, or 5 UTP cabling.
2. R/O = Read-Only.
3. 100BASE-TX allows users to run 100BASE-T over two pairs of Category 5 UTP and Type 1 STP.
4. LH = Latching High. When a fault condition occurs (remote fault or jabber), the bit is set. The bit remains set until the PHY status register is read via the MIIM or the PHY is reset, at which time the bit is cleared.
5. LL = Latching Low. When the link is not valid, the PHY clears the bit. It remains cleared until the PHY status register is read via the MIIM. The bit is set when the PHY has determined that a valid link has been established.

3.2.3.3 Scan Operation

When the host asserts the SCAN signal, the MIIM core performs a continuous read operation of the PHY register specified by the FIAD[4:0] and RGAD[4:0] signals. The scan operation can be used to poll the PHY Status Register (address 0x001), which allows the host to monitor the Link Status bit. The MIILF signal output reflects the state of the Link Status bit in the PHY status register.

During a scan operation, the MIIM core keeps the BUSY signal asserted until the last read is finished. If the SCAN signal is deasserted before the end of the current read operation, the MIIM core completes the current read operation and then deasserts the BUSY signal. For every read performed during the scan operation, the PRSD[15:0] signals are also updated. During the scan operation, the NVALID signal should be used to qualify the validity of the PRSD[15:0] and MIILF signals.

When the SCAN signal is asserted to perform a continuous scan, the RSTAT signal must be deasserted.

Chapter 4
Functional Timing

This chapter examines various E-110 functional timing scenarios. The key timing relationships are discussed. For details of the operation of each of the signals discussed, refer to Chapter 2, “Signal Descriptions.” This chapter contains the following sections:

- Section 4.1, “MAC Functional Timing”
- Section 4.2, “MIIM Core Functional Timing”
- Section 4.3, “Transmit Collision Functional Timing”

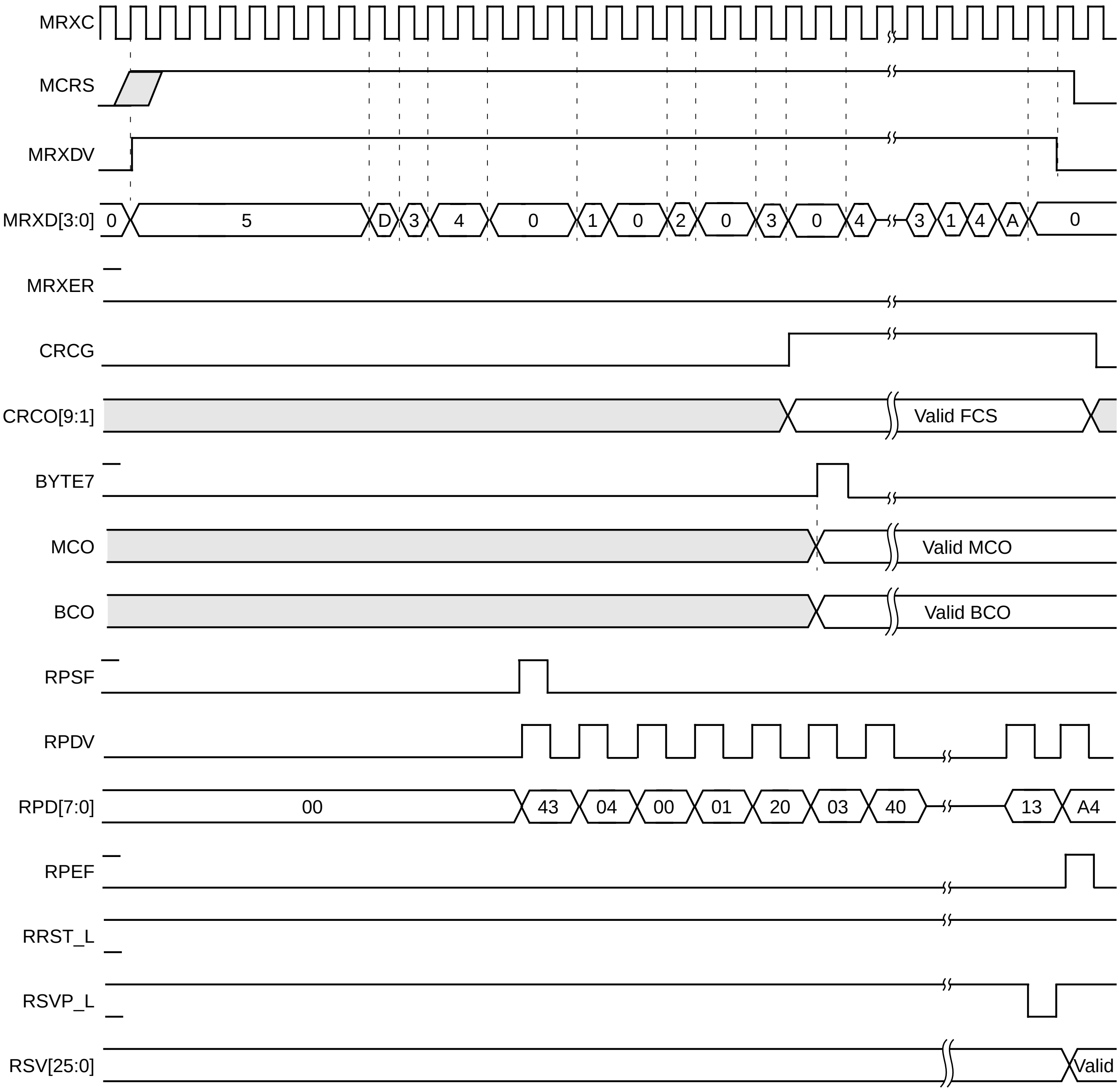
4.1 MAC Functional Timing

The following subsections describe the functional timing for MAC operations, including MAC receive and transmit packet timing.

4.1.1 MAC Receive Packet Timing

The timing for a typical receive packet is shown in Figure 4.1. Notice that the PHY asserts the MCRS and MRXDV signals at the beginning of packet reception. The PHY asserts MCRS asynchronously as soon as it detects a non-idle medium. The PHY also asserts MRXDV synchronously to the rising edge of MRXC when it has decoded a valid preamble.

Figure 4.1 E-110 MAC Receive Packet Timing

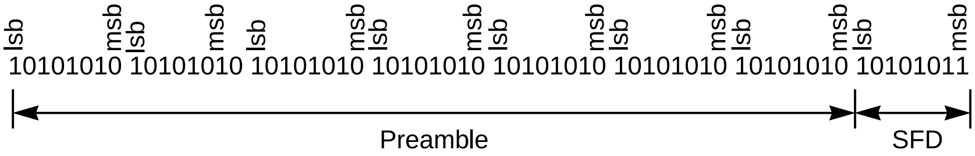


Note: RSVP_L is asserted one clock before RPEF.

4.1.1.1 Preamble

Every packet is preceded by a preamble consisting of seven octets followed by a single SFD octet. The preamble and SFD are shown in Figure 4.2.

Figure 4.2 Packet Preamble and SFD



In Figure 4.2, the preamble and SFD are shown in the serial bit order of reception. For each octet, the leftmost bit in each octet value represents the least significant bit of the octet and the rightmost bit represents the most significant bit of the octet. The preamble consists, then, of seven octets of 0x55. The SFD octet (0xD5) indicates the start of frame and follows the preamble. In Figure 4.1, the MRXD[3:0] data changes in the beginning from 0x0 to 0x5 to 0xD, which represents the transition from idle to preamble to SFD.

4.1.1.2 Nibble and Byte Assembly

Data is passed to the E-110 MAC from the PHY as nibbles on the MRXD[3:0] lines. Because the data is received least significant bit first, the low nibble is received, then the high nibble. Figure 4.1 shows that the PHY transfers the MRXD[3:0] nibbles received after the SFD to the E-110 MAC in the following order: 0x3, 0x4, 0x4, 0x0, 0x0, 0x0, 0x1, 0x0, 0x0, 0x2, 0x0, 0x0, 0x3, 0x0, 0x0, 0x4, and so on. The MAC then assembles the nibbles into bytes with the nibbles in reverse position, so that the high and low nibbles are in the proper order for the host on the RPD[7:0] signals. The RPD[7:0] data in Figure 4.1 shows that the MAC sends the bytes to the host in the following order: 0x43, 0x04, 0x00, 0x01, 0x20, 0x03, 0x40, and so on.

A packet transmission to the receive control logic from the receive function begins when the receive function asserts the RPSF and RPDV signals along with the first byte of received packet data after removing the preamble and SFD. For subsequent data bytes, the receive function asserts only the RPDV signal until the last byte, when it asserts both the RPDV and RPEF signals.

4.1.1.3 FCS Operation

The CRCO[9:1] lines reflect the state of the receive function FCS register after the first 48 bits (six bytes) of the receive packet. The CRCG signal, when asserted, indicates that the CRCO[9:1] signals are valid. The CRCO[9:1] signals contain the nine-bit hash function computed from the destination address. The nine-bit hash function is generated after the final destination address bit is received.

The nine-bit value can be used as an index into an external multicast filter hash table to determine if the frame should be received. External logic can decide to either accept or reject the incoming frame.

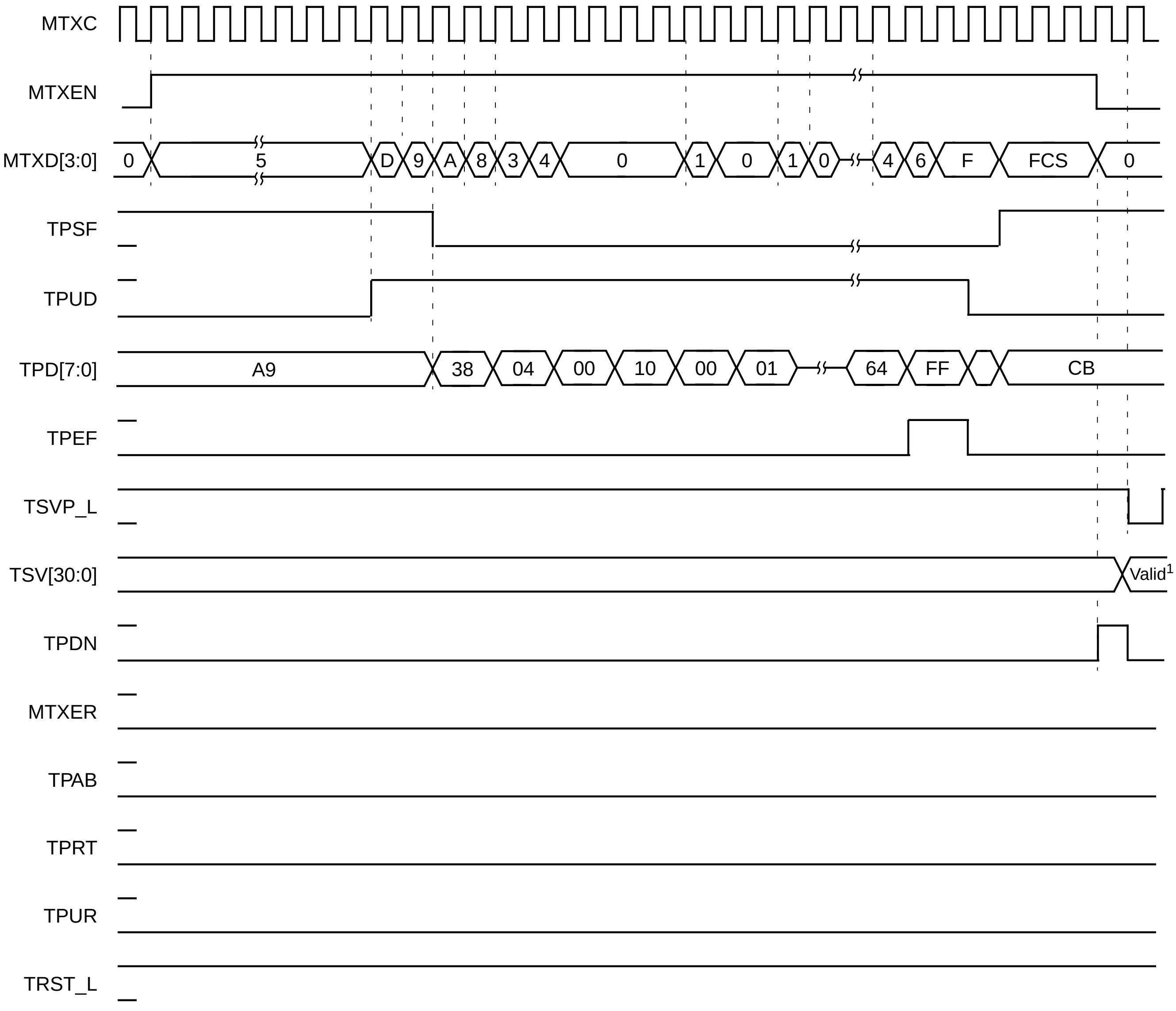
4.1.1.4 End of Frame

When asserted, the RPEF signal marks the last byte of the receive packet. At the end of the packet, the E-110 MAC asserts RSVP_L, an active-LOW pulse that indicates the availability of a new receive statistics vector on the RSV[25:0] lines.

4.1.2 MAC Transmit Packet Timing

The timing for a typical transmit packet is shown in Figure 4.3.

Figure 4.3 E-110 MAC Transmit Packet Timing



1. Transmit status vector valid.

4.1.2.1 Start of Transmission

When the MAC is ready to begin a packet transmission, it asserts the MTXEN signal to the PHY to indicate that the MAC is presenting nibbles on the MII for transmission.

4.1.2.2 Preamble

In the example shown in Figure 4.3, the MAC begins sending the preamble to the PHY. The preamble consists of the value 0x5 on the

MTXD[3:0] lines (as soon as MTXEN is asserted). As an option, the host may assert the NOPRE signal to instruct the MAC not to send a preamble, in which case the host itself must supply the preamble. After sending 15 nibbles of 0x5, the MAC sends a 0xD, which is the SFD. The preamble and SFD add up to 16 nibbles (8 bytes).

4.1.2.3 Nibble and Byte Assembly

Data is passed from the E-110 MAC to the PHY as nibbles on the MTXD[3:0] lines. The TPD[7:0] data in Figure 4.3 shows that the MAC receives the bytes from the host in the following order: 0xA9, 0x38, 0x04, 0x00, 0x10, 0x00, 0x01, and so on. Because the data is sent to the PHY least significant bit first, the low nibble is sent first, followed by the high nibble. The MAC must assemble the host bytes into nibbles with the nibbles in reverse position, so that the low and high nibbles are in the proper serial bit order for transmission to the PHY on TXD[3:0]. Figure 4.3 shows that the MAC transfers the MTXD[3:0] nibbles to the PHY after the SFD in the following order: 0x9, 0xA, 0x8, 0x3, 0x4, 0x0, 0x0, 0x0, 0x0, 0x1, 0x0, 0x0, 0x1, 0x0, and so on.

MTXD[3:0] nibbles are valid only when the MTXEN signal is asserted. The transmit function supplies nibbles of zero when MTXEN is not asserted. When MTXEN is asserted, the transmit function supplies preamble, SFD, transmit data, FCS, jam, or pad nibbles.

4.1.2.4 End of Frame

When asserted, the TPEF signal marks the last byte of the transmit packet and must be valid for two transmit clock periods. The MAC appends the FCS bytes to the packet data from the host and then asserts the TPDN signal to the host, indicating successful packet completion. The MAC then asserts the TSVP_L signal to indicate the availability of a new transmit statistics vector (TSV[30:0]).

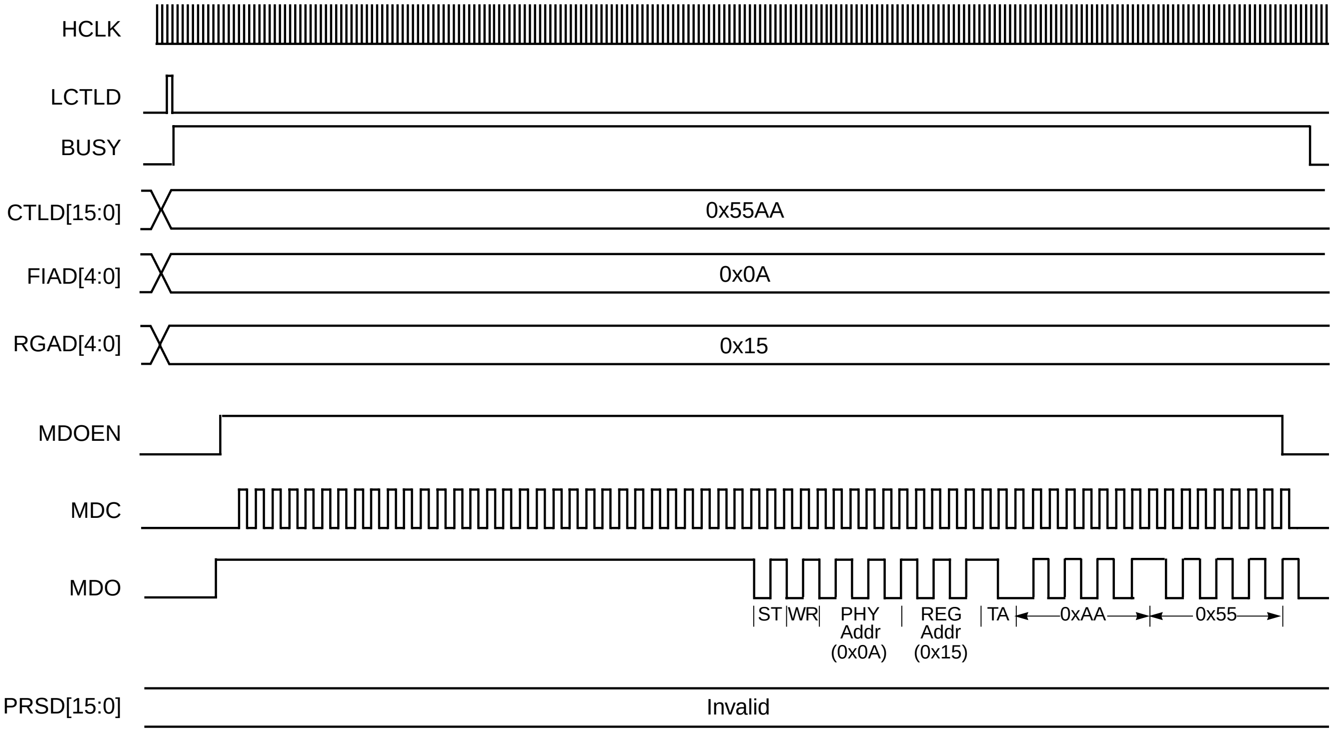
4.2 MIIM Core Functional Timing

The following subsections describe the functional timing for MIIM operations, including MIIM write and read timing.

4.2.1 MIIM Write Operation

The timing for an MIIM write operation from the MIIM core to a PHY is shown in Figure 4.4.

Figure 4.4 MIIM Write Operation



The timing shows the sequence of events to write to a PHY control register. The values used for addresses and data are illustrative only.

The host asserts the LCTLD signal to initiate the write operation. LCTLD indicates that the host data lines (CTLD[15:0]), PHY address (FIAD[4:0]), and PHY register address (RGAD[4:0]) are valid. In the example, CTLD[15:0] contains 0x55AA, which is the data being written to the PHY control register. FIAD[4:0] contains 0x0A, and RGAD[4:0] contains 0x15.

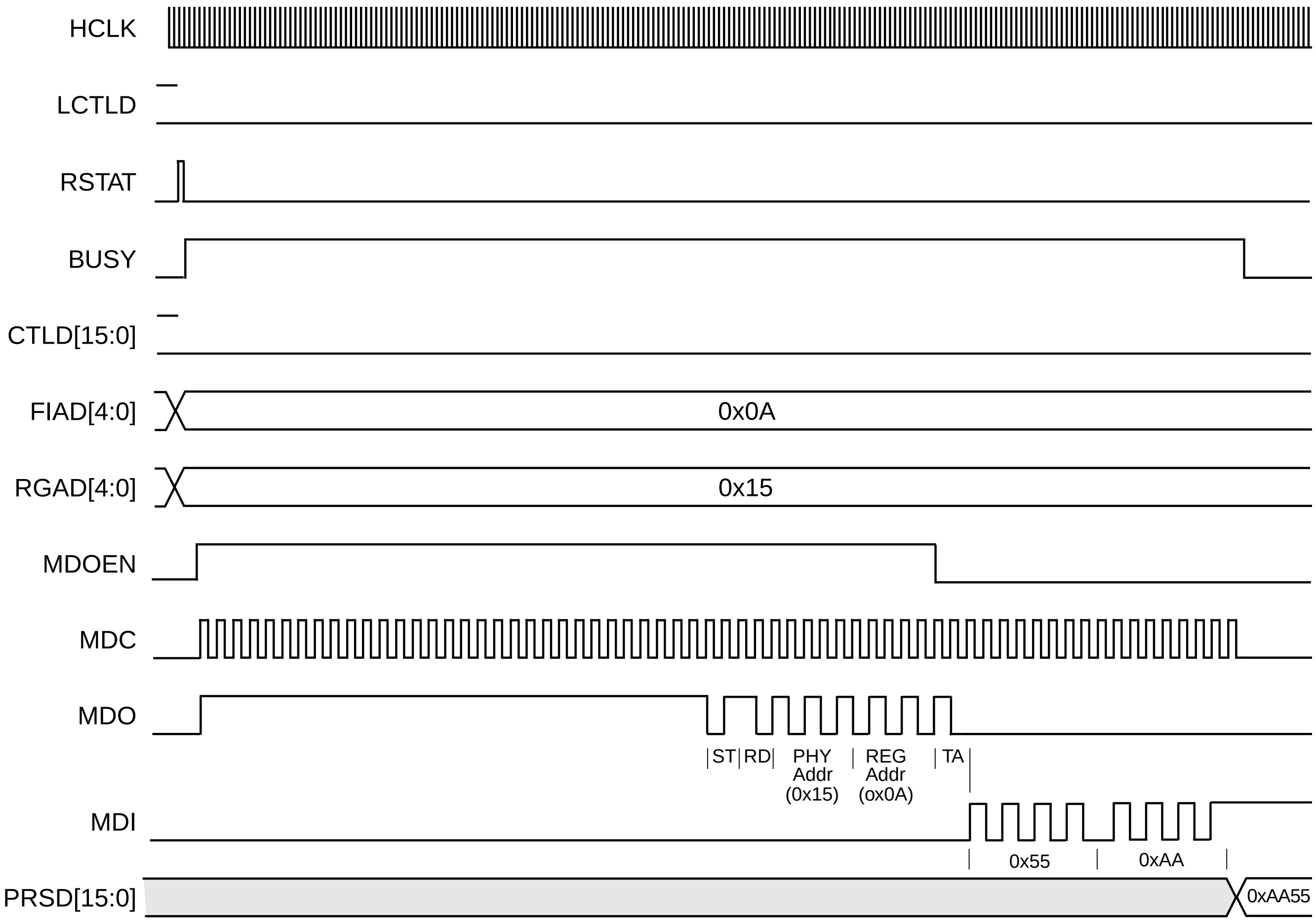
As soon as the host loads the data with the LCTLD signal, the MIIM core asserts the BUSY signal to the host, indicating that a PHY write operation is in process. The core asserts the MDOEN signal to enable the output data line, MDO, to the PHY. The core then clocks out data to the PHY on each rising edge of MDC in accordance with the MIIM frame format described in the IEEE 803.2u specification. The frame structure is briefly summarized below and is illustrated in Figure 4.4.

On MDO, the MIIM core sends a preamble of 32 contiguous logic one bits, followed by a start of frame pattern (ST), which is a zero-one combination. The next two bits, a zero-one sequence (WR) specifies that the core is performing a write operation. The PHY address, 0x0A, is next, followed by the PHY register number, 0x15. The next field is the two-bit turnaround (TA) field. For a write operation, the field is defined to be a one-zero sequence. Finally, the core sends the data, 0x55AA.

4.2.2 MAC MIIM Read Operation

The timing for an MIIM read operation from a PHY to the MIIM core is shown in Figure 4.5.

Figure 4.5 MIIM Read Operation



The timing shows the sequence of events to read from a PHY status register. The values used for data and address are illustrative only, and the values used in this read example bear no relation to those used in the write example.

The host asserts the RSTAT signal to initiate the read operation. RSTAT indicates that the host PHY address (FIAD[4:0]) lines and PHY register address lines (RGAD[4:0]) are valid. In the example, FIAD[4:0] contains 0x15, and RGAD[4:0] contains 0x0A.

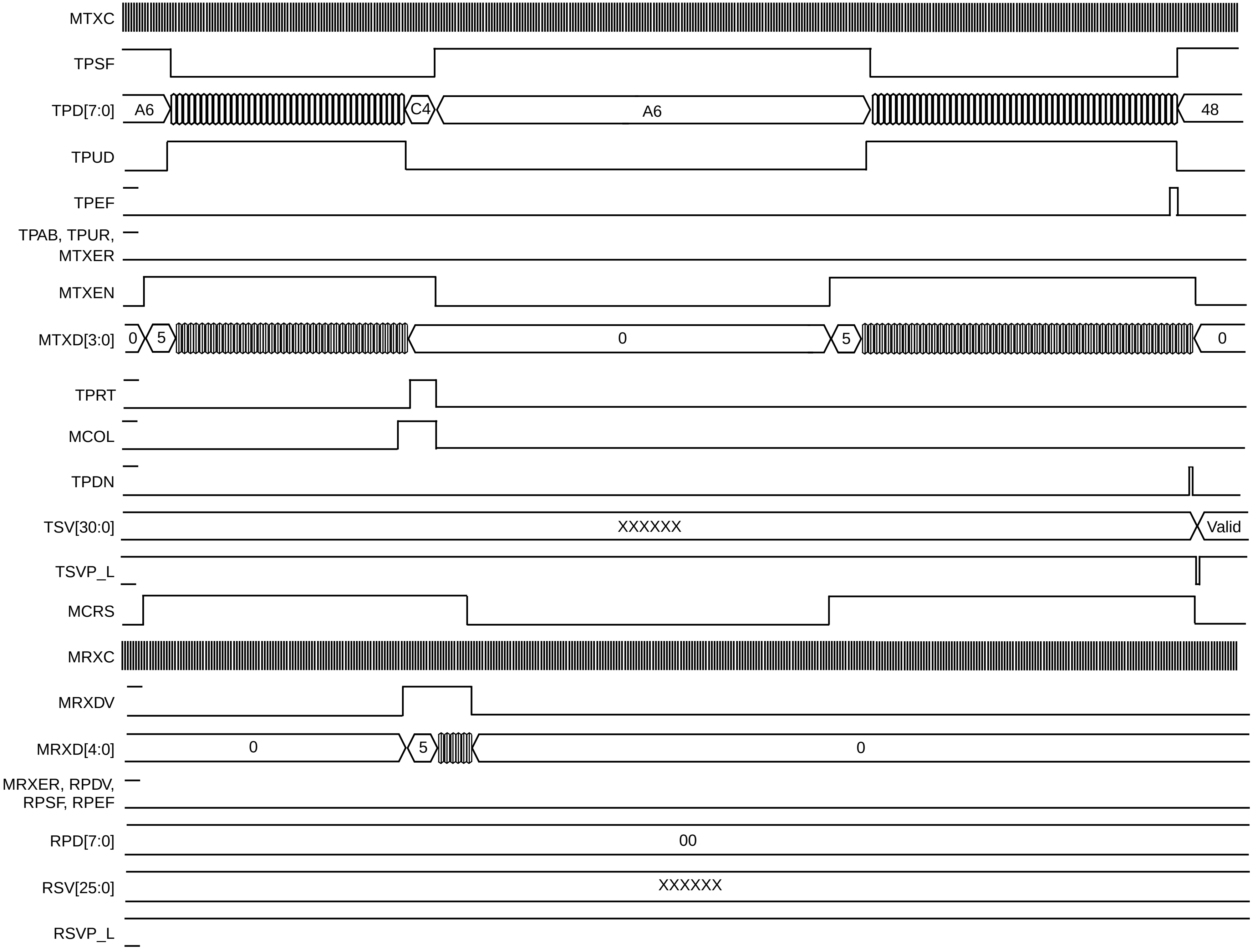
As soon as the host asserts RSTAT, the MIIM core asserts the BUSY signal to the host, indicating that a PHY read status operation is in process. The core asserts the MDOEN signal to enable the output data line, MDO, to the PHY. The core then clocks out data to the PHY on MDO and clocks in status from the PHY on MDI on each rising edge of MDC in accordance with the MIIM frame format described in the IEEE 803.2u specification.

On MDO, the core sends a preamble of 32 contiguous logic one bits, followed by a start of frame pattern (ST), which is a zero-one combination. The next two bits, a one-zero sequence (RD), specify that the MIIM core is performing a read operation. The PHY address, 0x15, is next, followed by the PHY register number, (0x0A). The next field is the two-bit turnaround (TA) field. For a read operation, the field is defined to be a one-bit period in which the PHY remains in a high-impedance state followed by a one-bit period during which the PHY drives a zero on the MDO line. Finally, the core deasserts the MDOEN signal, which enables the MDI input and the PHY sends the status, 0xAA55, to the core on MDI. The core deasserts the BUSY signal at the end of the sequence, at which time it drives valid data on the PRSD[15:0] lines to the host.

4.3 Transmit Collision Functional Timing

Figure 4.6 is a functional timing diagram for transmit collisions. For a description of the transmit collision operation, see Section 3.2.1.1, “MAC Transmit Function,” on page 3-5.

Figure 4.6 Transmit Collision Functional Timing



Chapter 5
Specifications

This chapter provides specifications for the E-110 core, including the AC timing and a pin summary.

This chapter has three sections:

- Section 5.1, “Derivation of AC Timing and Loading”
- Section 5.2, “MAC Core AC Timing”
- Section 5.3, “MAC Core Pin Summary”
- Section 5.4, “MAC Control Module Core AC Timing”
- Section 5.5, “MAC Control Module Core Pin Summary”
- Section 5.6, “MIIM Core AC Timing”
- Section 5.7, “MIIM Core Pin Summary”

5.1 Derivation of AC Timing and Loading

Delay predictor software is included with every core LSI Logic delivers for incorporation into an ASIC. The software generates an input loading report so you can plan for buffer strengths that drive core inputs.

A ramp time violation report is generated when you integrate the core into the rest of your logic and run other simulation software. The report indicates if a core output is too heavily loaded. Adjust buffering, wire length, and other parameters to eliminate the violation.

There are no specific numbers in this chapter for AC timing and loading, because the numbers depend on the technology used and the design layout.

5.2 MAC Core AC Timing

This section provides a list of signals that must operate properly with respect to the MTXC, MRXC, and HCLK signals. The relationship between the signals is depicted in Figure 5.1.

The numbers in Figure 5.1 refer to the AC timing parameters listed in the first column of Table 5.1. The core has been verified under worst-case process voltage and ambient temperature.

Figure 5.1 MAC Core Setup, Hold, and Delay Timing Diagram

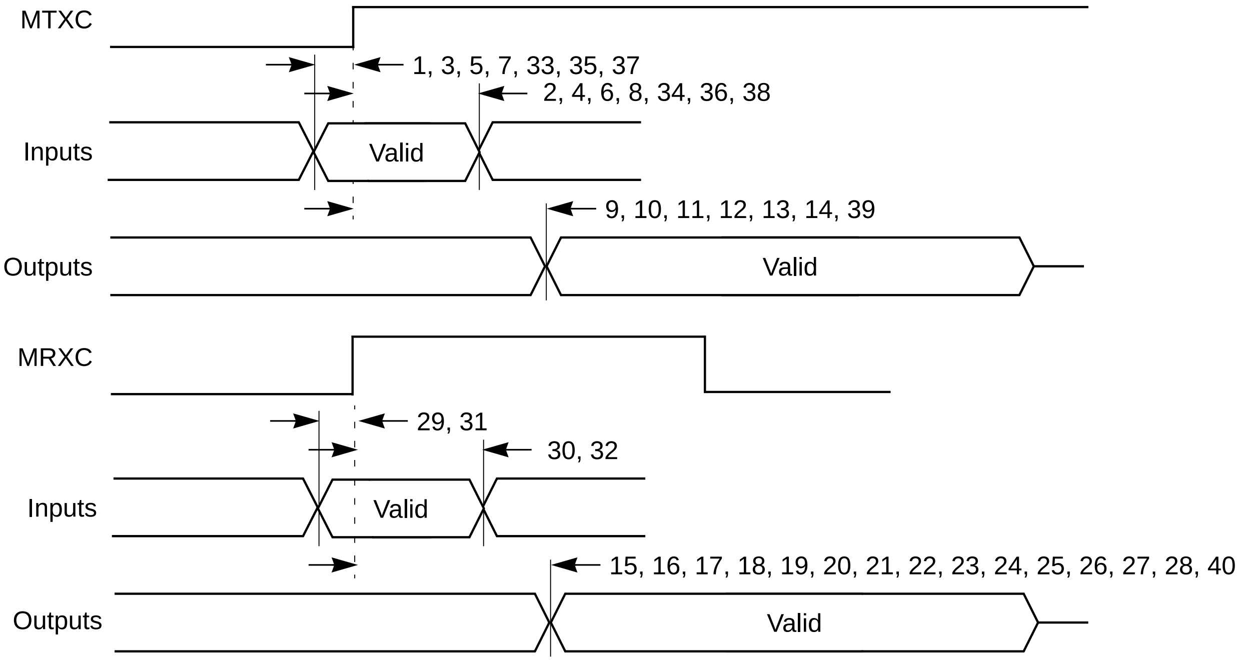


Table 5.1 MAC Core AC Timing Parameters

Parameter	Description	Parameter	Description
1.	TPSF setup to MTXC	2.	TPSF hold time after MTXC
3.	TPEF setup to MTXC	4.	TPEF hold time after MTXC
5.	TPUR setup to MTXC	6.	TPUR hold time after MTXC
7.	TPD[7:0] setup to MTXC	8.	TPD[7:0] hold time after MTXC
9.	TPUD delay from MTXC	10.	TPDN delay from MTXC
11.	TPRT delay from MTXC	12.	TPAB delay from MTXC
13.	TSVP_L delay from MTXC	14.	TSV[30:0] delay from MTXC
15.	RPD[7:0] delay from MRXC	16.	RPDV delay from MRXC
17.	RPSF delay from MRXC	18.	RPEF delay from MRXC
19.	CRCO[9:1] delay from MRXC	20.	CRCG delay from MRXC
21.	BCO delay from MRXC	22.	MCO delay from MRXC
23.	BYTE7 delay from MRXC	24.	RSVP_L delay from MRXC
25.	RSV[25:0] delay from MRXC	26.	PRSD[15:0] delay from MRXC
27.	MIILF delay from MRXC	28.	NVALID delay from MRXC
29.	MRXD[3:0] setup to MRXC	30.	MRXD[3:0] hold time after MRXC
31.	MRXDV setup to MRXC	32.	MRXDV hold time after MRXC
33.	TBOFF_SEL setup before MTXC	34.	TBOFF_SEL hold after MTXC
35.	BKOFF_LIMIT[1:0] setup before MTXC	36.	BKOFF_LIMIT[1:0] hold after MTXC
37.	FLS_CRS setup before MTXC	38.	FLS_CRS hold after MTXC
39.	TRST_L delay from MTXC	40.	RRST_L delay from MRXC

5.3 MAC Core Pin Summary

Table 5.2 summarizes the E-110 core input and output signals. The table provides the signal names and types for both outputs and inputs.

Table 5.2 MAC Core Pin Summary

Mnemonic	Description	Type	Active
BCO	Broadcast Out	Output	HIGH
BKOFF_LIMIT[1:0]	Backoff Limit	Input	—
BYTE7	Byte 7	Output	HIGH
CRCEN	CRC Append Enable	Input	HIGH
CRCG	CRC Good	Output	HIGH
CRCO[9:1]	CRC Out	Output	—
FLS_CRS	False Carrier Sense (Backpressure Control)	Input	HIGH
FULLD	Full-Duplex Control	Input	HIGH
HRST_L	Host Reset	Input	LOW
HUGEN	Huge Packet Enable	Input	HIGH
HWD[10:0]	Host Write Data [10:0]	Input	—
IPGR1[6:0]	Non Back-to-Back IPG (Part 1)	Input	—
IPGR2[6:0]	Non Back-to Back IPG (Part 2)	Input	—
IPGT[6:0]	Back-to-Back IPG	Input	—
LRNG	Load Random Number Generator	Input	HIGH
MCO	Multicast Out	Output	HIGH
MCOL	Collision Detected	Input	HIGH
MCRS	Carrier Sense	Input	HIGH
MRXC	Receive Nibble or Symbol Clock	Input	HIGH
MRXD[3:0]	Receive Nibble Data	Input	—
(Sheet 1 of 3)			

Table 5.2 MAC Core Pin Summary (Cont.)

Mnemonic	Description	Type	Active
MRXDV	Receive Data Valid	Input	HIGH
MRXER	Receive Error	Input	HIGH
MTXC	Transmit Nibble or Symbol Clock	Input	HIGH
MTXD[3:0]	Transmit Nibble Data	Output	—
MTXEN	Transmit Enable	Output	HIGH
MTXER	Transmit Coding Error	Output	HIGH
NOPRE	No Preamble	Input	HIGH
PADEN	Pad Enable	Input	HIGH
RPD[7:0]	Receive Packet Data	Output	—
RPDV	Receive Packet Data Valid	Output	HIGH
RPEF	Receive Packet End Of Frame	Output	HIGH
RPSF	Receive Packet Start Of Frame	Output	HIGH
RRST_L	Receive Reset	Output	LOW
RSTAT	Request MII Status	Input	HIGH
RSV[25:0]	Receive Statistics Vector	Output	HIGH
RSVP_L	Receive Statistics Vector Pulse	Output	LOW
RTRYL	Retry Late Collision	Input	HIGH
SCAN	Status Read Scan	Input	HIGH
TBOFF_SEL	Stop Backoff	Input	HIGH
TEST_SE	Scan Enable	Input	HIGH
TEST_SI	Scan Input	Input	—
TEST_SO	Scan Output	Output	—
TMODE	Test Mode	Input	HIGH
TPAB	Transmit Packet Abort	Output	HIGH
(Sheet 2 of 3)			

Table 5.2 MAC Core Pin Summary (Cont.)

Mnemonic	Description	Type	Active
TPD[7:0]	Transmit Packet Data	Input	—
TPDN	Transmit Packet Done	Output	HIGH
TPEF	Transmit Packet End of Frame	Input	HIGH
TPRT	Transmit Packet Retry	Output	HIGH
TPSF	Transmit Packet Start Of Frame	Input	HIGH
TPUD	Transmit Packet Data Used	Output	HIGH
TPUR	Transmit Packet Data Underrun	Input	HIGH
TRST_L	Transmit Reset	Output	LOW
TSV[30:0]	Transmit Statistics Vector	Output	—
TSVP_L	Transmit Statistics Vector Pulse	Output	LOW
(Sheet 3 of 3)			

5.4 MAC Control Module Core AC Timing

This section gives a list of the signals that must operate properly with respect to the MTXC and MRXC signals. The relationship between the signals is depicted in Figure 5.2. The numbers in Figure 5.2 refer to the AC timing parameters listed in the first column of Table 5.3. The MAC control module core has been verified under worst-case process, voltage, and ambient temperature.

Figure 5.2 MAC Control Module Setup, Hold, and Delay Timing Diagram

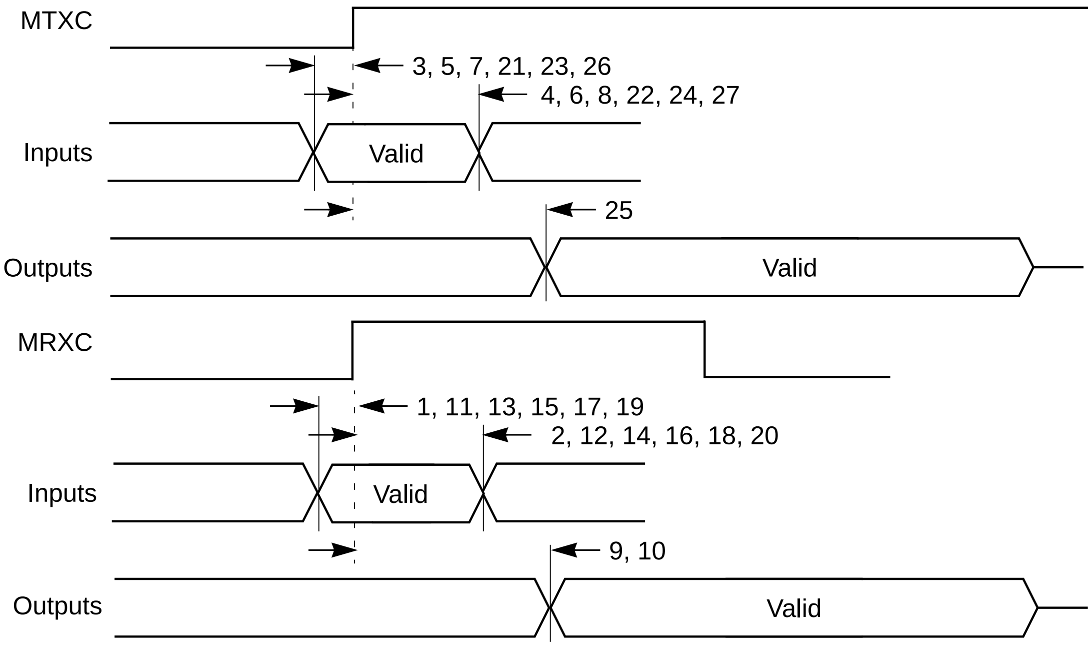


Table 5.3 MAC Control Module AC Timing Parameters

Parameter	Description	Parameter	Description
1.	ADDR_MATCH setup to MRXC	2.	ADDR_MATCH hold time after MRXC
3.	CTLP setup to MTXC	4.	CTLP hold time after MTXC
5.	FLOWCON_EN setup to MTXC	6.	FLOWCON_EN hold time after MTXC
7.	HST_TPSF setup to MTXC	8.	HST_TPSF hold time after MTXC
9.	PAUSE_ACTIVE delay from MRXC	10.	RCVNG_PAUSE_FRAME delay from MRXC
11.	RPD[7:0] setup to MRXC	12.	RPD[7:0] hold time after MRXC
13.	RPDV setup to MRXC	14.	RPDV hold time after MRXC
15.	RPEF setup to MRXC	16.	RPEF hold time after MRXC
17.	RPSF setup to MRXC	18.	RPSF hold time after MRXC
19.	RRST_L setup to MRXC	20.	RRST_L hold time after MRXC
(Sheet 1 of 2)			

Table 5.3 MAC Control Module AC Timing Parameters (Cont.)

Parameter	Description	Parameter	Description
21.	TPAB setup to MTXC	22.	TPAB hold time after MTXC
23.	TPDN setup to MTXC	24.	TPDN hold time after MTXC
25.	TPSF delay after MTXC	26.	TRST_L setup to MTXC
27.	TRST_L hold time after MTXC		
(Sheet 2 of 2)			

5.5 MAC Control Module Core Pin Summary

Table 5.4 summarizes the MAC control module core signals.

Table 5.4 MAC Control Module Pin Summary

Mnemonic	Description	Type	Active
ADDR_MATCH	Unicast Address Match	Input	HIGH
CTLP	Host Control Packet Transmit Request	Input	HIGH
FLOWCON_EN	Flow Control Enable	Input	HIGH
HST_TPSF	Host Packet Transmit Request	Input	HIGH
MRXC	Receive Nibble or Symbol Clock	Input	—
MTXC	Transmit Nibble or Symbol Clock	Input	—
PAUSE_ACTIVE	Pause in Progress	Output	HIGH
PAUSE_MIRROR[23:0]	Pause Mirror Counter Value	Output	—
PAUSE_RSVP_L	Comprehensive Receive Statistics Vector Pulse	Output	LOW
PAUSE_TIME_IN[15:0]	Transmitted Pause Time	Input	—
RCV_LOAD_DLY[3:0]	Receiver Delay in Loading Transmitted Pause	Input	—
RCVNG_PAUSE_FRAME	Receiving Pause Frame	Output	HIGH
RPD[7:0]	Receive Packet Data	Input	—
(Sheet 1 of 2)			

Table 5.4 MAC Control Module Pin Summary (Cont.)

Mnemonic	Description	Type	Active
RPDV	Receive Packet Data Valid	Input	HIGH
RSV_GOOD	Received Statistics Vector Good	Input	HIGH
RSVD_PAUSEF	Received Pause Frame	Output	HIGH
RPEF	Receive Packet End of Frame	Input	HIGH
RPSF	Receive Packet Start of Frame	Output	HIGH
RRST_L	Receive Reset	Input	LOW
TPAB	Transmit Packet Abort	Input	HIGH
TPDN	Transmit Packet Done	Input	HIGH
TPSF	Transmit Packet Start of Frame	Output	HIGH
TRST_L	Transmit Reset	Input	LOW
XMT_PAUSEF	Transmitted Pause Frame	Output	HIGH
UNSUPP_OPCODE	Unsupported Opcode	Output	HIGH
(Sheet 2 of 2)			

5.6 MIIM Core AC Timing

This section gives a list of the signals that must operate properly with respect to the HCLK signal. The relationship between the signals is depicted in Figure 5.3. The numbers in Figure 5.3 refer to the AC timing parameters listed in the first column of Table 5.5. The MIIM core has been verified under worst-case process, voltage, and ambient temperature.

Figure 5.3 MIIM Core Setup, Hold, and Delay Timing Diagram

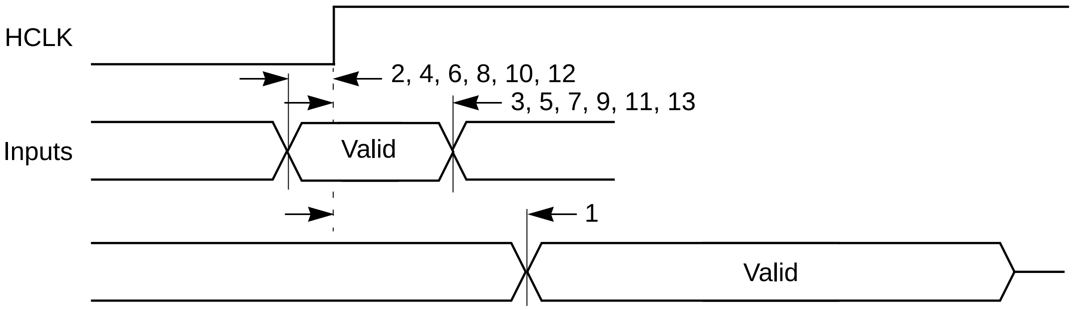


Table 5.5 MIIM Core AC Timing Parameters

Parameter	Description	Parameter	Description
1.	BUSY delay from HCLK	2.	CTLD[15:0] setup to HCLK
3.	CTLD[15:0] hold time after HCLK	4.	FIAD[4:0] setup to HCLK
5.	FIAD[4:0] hold time after HCLK	6.	RGAD[4:0] setup to HCLK
7.	RGAD[4:0] hold time after HCLK	8.	LCTLD setup to HCLK
9.	LCTLD hold time after HCLK	10.	RSTAT setup to HCLK
11.	RSTAT hold time after HCLK	12.	SCAN setup to HCLK
13.	SCAN hold time after HCLK		

5.7 MIIM Core Pin Summary

Table 5.6 summarizes the MIIM core input and output signals. The table provides the signal names and types for both outputs and inputs.

Table 5.6 MIIM Core Pin Summary

Mnemonic	Description	Type	Active
BUSY	Busy	Output	HIGH
CLKS	Clock Select	Input	HIGH
CTLD[15:0]	Control Data	Input	—
FIAD[4:0]	PHY Address[4:0]	Input	—
HCLK	Host Clock	Input	HIGH
HRST_L	Host Reset	Input	LOW
LCTLD	Load Control Data	Input	HIGH
MDC	MIIM Data Clock	Output	HIGH
MDI	MIIM Data In	Input	—
MDO	MIIM Data Out	Output	—
MDOEN	MIIM Data 3-State Enable	Output	HIGH
MIILF	MII Link Failure	Output	HIGH
NVALID	Invalid	Output	HIGH
PRSD[15:0]	PHY Read Status Data	Output	—
RGAD[4:0]	Register Address[4:0]	Input	—
RSTAT	Request MII Status	Input	HIGH
SCAN	Status Read Scan	Input	HIGH

Appendix A
Glossary

Address Filtering (Individual, Multicast, Promiscuous)

Address filtering is the process the E-110 core performs to match a destination address within a received frame with an address derived at the receiving station. There are three common types of address filtering:

- Individual—the first bit of an individual address is a zero. The incoming destination address is compared with the data from the individual address pins. When all 48 bits match and the receiver is enabled, the address filter passes.
- Multicast—the first bit of an multicast address is a one. When the destination address bits that are received in the frame contain a multicast address, the E-110 core uses its built-in FCS generator to compute a nine-bit polynomial (the nine most significant bits of the 32-bit FCS generator) from the incoming address. The value of this polynomial can be used as an index into an external multicast filter hash table. External logic can decide to either accept or reject the incoming frame. The external logic enables the multicast address filter.
- Promiscuous—when the external logic asserts the individual address promiscuous mode enable pin (RPMEN) HIGH and the destination address is an individual address, the address filter always passes, and the incoming frame is received.

Attachment Unit Interface (AUI)

The AUI is the interface between the medium attachment unit (MAU) and the data terminal equipment (DTE) device or repeater. A typical AUI interface consists of a 15-pin “D” connector.

Backoff

In IEEE 802.3 networks, backoff occurs when two or more nodes attempt a transmission and collide. The function of stopping transmission and waiting a specified random time before retrying the transmission is considered backoff. In IEEE 802.3 networks a “truncated binary exponential backoff” algorithm is employed.

Broadcast Packet

A broadcast packet has the destination address field set to all ones, which indicates it is being sent to all destinations on the network.

Bridge

A bridge connects distinct segments of an Ethernet LAN (usually referring to a physical length of wire) and transmit traffic between them. A bridge allows you to extend the maximum size of the network while still not exceeding the maximum wire length, attached device count, or number of repeaters for a network segment.

Collision

When an Ethernet station is operating in the AUI mode, it can sense a change in the energy level of the communication channel and interpret the phenomenon as a collision. A collision is caused by two stations attempting to transmit at the same time.

Collision Detection

Collision detection is the ability of a transmitting node on an Ethernet LAN to sense a change in the energy level of the channel and to interpret the phenomenon as a collision.

CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

A LAN protocol access method where the nodes are attached to a cable. When a node transmits data onto the network and raises the carrier, the remaining nodes detect the carrier (Carrier Sense) and “listen” for the information to detect if it is intended for them. The nodes have network access (Multiple Access) and can send if no other transmission is taking place. If two nodes attempt to send simultaneously, a collision takes place (Collision Detection) and both nodes must retry at random intervals.

Deference

Deference (or deferral length) is the time the MAC transmit function waits for the network to be free before it can transmit. See also *Excess Deferral*.

Dribble Nibble

A nibble that is “extra” and occurs when, at the end of a frame, less than a complete byte is received. It is possible to have one dribble nibble at the end of a frame if a complete byte is not received.

End of Frame Delimiter

Three consecutive ones at the end of a frame that are not Manchester-encoded; that is, they are non-return-to-zero bits that do not contain a transition in the middle of the bit time. The purpose of the end of frame delimiter is to cause the phase-lock loop at the receiving station to lose lock so the host can be notified that the frame has ended.

When a receiving node detects a bit that is not Manchester-encoded, it considers the previous four bytes to be the FCS.

Ethernet LAN

A branching broadcast communications system for carrying digital data packets among locally distributed computing stations. Ethernet is a 10-Mbit/s baseband, local area network that has evolved into the IEEE 802.3 specification and is defined by a data-link protocol that specifies how data is placed on and retrieved from a common transmission medium. Ethernet is used as the underlying transport vehicle by several upper level protocols, including TCP/IP and Xerox Network System (XNS). See *IEEE 802.3/IEEE 802.3u*.

Excess Deferral

The time period when the media is busy longer than twice the maximum frame size. When HUGEN is deasserted, the maximum frame size is 1518 bytes, so excess deferral in this case is:

1518 bytes x 8 bits/byte x 2 = 24,288 bits (242.88 µs at 100 Mbit/s or 2.4288 ms at 10 Mbit/s)

When HUGEN is asserted, the maximum frame size is 32 Kbytes, so excess deferral in this case is:

32 Kbytes x 8 bits/byte x 2 = 524,288 bits (5242.88 µs at 100 Mbit/s or 52.43 ms at 10 Mbit/s)

Fast Ethernet

Compared to the 10-mbit/s specifications, the 100-Mbit/s Fast Ethernet system results in a factor of ten reduction in the bit time, which is the amount of time it takes to transmit a bit on the Ethernet channel. The result is a tenfold increase in the speed of the packets over the media system. However, the other important aspects of the Ethernet system, including the frame format, the amount of data a frame may carry, and the media access control mechanism, are all unchanged.

The Fast Ethernet specifications include mechanisms for auto-negotiation of the media speed. This makes it possible for vendors to provide dual-speed Ethernet interfaces that can be installed and run at either 10 Mbit/s or 100 Mbit/s automatically.

There are three media varieties that have been specified for transmitting 100-Mbit/s Ethernet signals:

- 100BASE-TX
- 100BASE-FX
- 100BASE-T4

The identifiers include three pieces of information. The first item, “100,” stands for the media speed of 100 Mbit/s. “BASE” stands for “baseband,” which is a type of signaling. Baseband signaling simply means that Ethernet signals are the only signals carried over the media system. The third part of the identifier provides an indication of the segment type. The “T4” segment type is a twisted-pair segment that uses four pairs of telephone-grade twisted-pair wire. The “TX” segment type is a twisted-pair segment that uses two pairs of wires and is based on the data grade twisted-pair physical medium standard developed by ANSI. The “FX” segment type is a fiber optic link segment based on the fiber optic physical medium standard developed by ANSI and that uses two strands of fiber cable. The TX and FX medium standards are collectively known as 100BASE-X.

The 100BASE-TX and 100BASE-FX media standards used in Fast Ethernet are both adopted from physical media standards first developed by ANSI, the American National Standards Institute. The ANSI physical media standards were originally developed for the Fiber Distributed Data Interface (FDDI) LAN standard (ANSI standard X3T9.5), and are widely used in FDDI LANs.

Rather than “reinventing the wheel” when it came to signaling at 100 Mbit/s, the Fast Ethernet standard adapted these two ANSI media standards for use in the new Fast Ethernet medium specifications. The T4 standard was also provided to make it possible to use lower-quality twisted-pair wire for 100-Mbit/s Ethernet signals.

Frame

An Ethernet frame contains the following eight fields:

- Preamble
- Start of Frame Delimiter
- Destination Address
- Source Address
- Length
- Data
- Pad (if necessary)
- Frame Check Sequence

Frame Check Sequence (FCS)

The Ethernet transmit and receive algorithm uses the standard IEEE 802.3 four-byte FCS field to ensure data integrity. During data transmission, the E-110 core uses a 32-bit linear feedback shift register (LFSR) to compute the value that will be sent in the FCS field. The FCS value is a function of the content of the source address, destination address, length, data, and pad fields. On data reception, the E-110 core preloads the LFSR with all ones and updates this value with each byte received, including the received FCS. If the final value in the LFSR does not match a predetermined value after all bytes including the FCS are received, the E-110 core flags an FCS error.

Hub

The E-110 core operating in 10BASE-T mode uses a hub to concentrate connections to multiple terminals. Hubs come in various sizes, with 4-port, 8-port, and 12-port being the most common. The number of ports indicates the number of terminals that can be connected to the hub. Most hubs have built-in transceivers and network management features. Hubs are mainly used in star topology LANs. The E-110 core is specifically designed for multiple-core integration, especially for “hub-on-a-chip” implementations.

IEEE 802.3/IEEE 802.3u

IEEE 802.3 is a standard set by the IEEE for CSMA/CD 10 Mbit/s network protocol, which is a Physical Layer definition including specifications for cabling in addition to transmitting data and controlling cable access. IEEE 802.3u is an IEEE standard that describes the MAC, Physical Layer, MAU, and Repeater for 100-Mbit/s 100BASE-T operation, and is a supplement to the IEEE 802.3 specification. See *Ethernet LAN*.

Invalid Preamble

An invalid preamble has a content that is not 0x55 or contains the code 0x50555 in the last reception.

Jabber

A condition on an Ethernet network when a node transmits for longer than the specified time.

Jam

In IEEE 802.3 networks, when a collision occurs, the colliding nodes ensure that the collision is seen by the entire network by continuing to transmit for a minimum time during a collision. This occurrence is known as jamming.

Link Failure

In the twisted-pair mode, a link failure can be caused by a twisted-pair transceiver failure or by a broken twisted-pair receive cable. A link failure occurs when there are eight consecutive missing link integrity pulses.

Local Area Network (LAN)

A communications system linking computers together to form a network whose dimensions typically are less than five kilometers. Transmissions within a local area network generally are digital, carrying data among stations at rates usually above 1 Mbit/s. A LAN is an assembly of computing resources such as microcomputers (for example, PCs), printers, minicomputers and mainframes linked by a common transmission medium, including coaxial cable or twisted-pair wiring.

Long Event

A long event occurs if the MCRS signal is asserted for over 50,000 bit times if the HUGEN signal is not asserted and over 200,000 bit times if HUGEN is asserted. A long event is not detected in full duplex mode.

MII Management Frame Format

The MII Management frame format is used to write control information to the PHY and read status information from the PHY. It is described in detail in IEEE 802.3u, clause 22.2.4.4

Minimally Qualified Receive Event

An event occurring when the MAC receives at least one nibble of data beyond a valid preamble and SFD.

Packet

Data sent in the data field of an Ethernet frame.

Packet Receive Attempt

A packet receive attempt occurs when the MAC sees a valid preamble and a valid SFD. When the MAC sees these two events, it tries to receive a packet. Errors in the packet after the preamble and SFD may, however, cause the packet to be discarded.

Start of Frame Delimiter (SFD)

The eight-bit SFD field follows the seven-octet preamble at the beginning of a frame and contains the 0b10101011 bit pattern. This pattern allows synchronization of the data received in the rest of the frame.

Switch

Synonymous with *Bridge*.

Transceiver

A combined transmitter and receiver and is an essential element of all LAN networks. When an Ethernet LAN operates in the 10BASE-T mode with an MIIM interface to a PHY, a transceiver is generally used to send the MDO and MDI MAC signals over a single signal, MDIO, to the PHY. The MAC MDOEN signal controls the direction of data transfer through the transceiver.

Twisted-Pair

Twisted-pair wire is a cable comprised of two 18- to 24-AWG (American Wire Gauge) solid copper strands twisted around each other. The twisting provides a measure of protection from electromagnetic and radio-frequency interference (EMI/RFI). Two types are available: shielded and unshielded. The former is wrapped inside a metallic sheath that provides protection from EMI/RFI. The latter, also known as telephone wire, is covered with plastic or PVC, which provides no protection from EMI/RFI. In 10BASE-T terminology, a twisted-pair transmission system refers to the twisted-pair wire link and its two attached MAUs.

Underflow

An underflow, or underrun, condition occurs when the host does not supply new transmit packet bytes fast enough to keep up with the MAC transmit requirements during a packet transmission.

Customer Feedback

We would appreciate your feedback on this document. Please copy the following page, add your comments, and fax it to us at the number shown.

If appropriate, please also fax copies of any marked-up pages from this document.

Important: Please include your name, phone number, fax number, and company address so that we may contact you directly for clarification or additional information.

Thank you for your help in improving the quality of our documents.

Reader’s Comments

Fax your comments to:

LSI Logic Corporation
Technical Publications
M/S E-198
Fax: 408.433.4333

Please tell us how you rate this document: *E-110 Core Technical Manual.* Place a check mark in the appropriate blank for each category.

	Excellent	Good	Average	Fair	Poor
Completeness of information	____	____	____	____	____
Clarity of information	____	____	____	____	____
Ease of finding information	____	____	____	____	____
Technical content	____	____	____	____	____
Usefulness of examples and illustrations	____	____	____	____	____
Overall manual	____	____	____	____	____

What could we do to improve this document?

____

____

____

____

If you found errors in this document, please specify the error and page number. If appropriate, please fax a marked-up copy of the page(s).

____

____

____

Please complete the information below so that we may contact you directly for clarification or additional information.

Name ________________________ Date ____________

Telephone ____________________ Fax ____________________

Title ____

Department ____________________ Mail Stop __________

Company Name ________________________________

Street ____

City, State, Zip ______________________________

U.S. Distributors by State

A. E. Avnet Electronics
W. E. Wyle Electronics

Alabama
Huntsville
A. E. Tel: 205.837.8700
W. E. Tel: 800.964.9953

Alaska
A. E. Tel: 800.332.8638

Arizona
Phoenix
A. E. Tel: 602.736.7000
W. E. Tel: 800.528.4040
Tucson
A. E. Tel: 520.742.0515

California
Irvine
A. E. Tel: 949.789.4100
W. E. Tel: 800.626.9953
Los Angeles
A. E. Tel: 818.594.0404
W. E. Tel: 800.288.9953
Sacramento
A. E. Tel: 916.632.4500
W. E. Tel: 800.627.9953
San Diego
A. E. Tel: 619.571.7540
W. E. Tel: 800.829.9953
San Jose
A. E. Tel: 408.435.3500
Santa Clara
W. E. Tel: 800.866.9953
Woodland Hills
A. E. Tel: 818.594.0404

Colorado
Denver
A. E. Tel: 303.790.1662
W. E. Tel: 800.933.9953

Connecticut
Chesire
A. E. Tel: 203.271.5700
Wallingford
W. E. Tel: 800.605.9953

Delaware
North/South
A. E. Tel: 800.526.4812
Tel: 800.638.5988

Florida
Fort Lauderdale
A. E. Tel: 305.484.5482
W. E. Tel: 800.568.9953
Orlando
A. E. Tel: 407.657.3300
W. E. Tel: 407.740.7450
Tampa
W. E. Tel: 800.395.9953
St. Petersburg
A. E. Tel: 813.507.5000

Georgia
Atlanta
A. E. Tel: 770.623.4400
W. E. Tel: 800.876.9953

Hawaii
A. E. Tel: 800.851.2282

Idaho
A. E. Tel: 801.266.2022

Illinois
North/South
A. E. Tel: 847.797.7300
Tel: 314.291.5350
Chicago
W. E. Tel: 800.853.9953

Indiana
Indianapolis
A. E. Tel: 317.575.3500
W. E. Tel: 888.358.9953

Iowa
Cedar Rapids
A. E. Tel: 319.393.0033

Kansas
Kansas City
A. E. Tel: 913.663.7900

Kentucky
Central/Northern/ Western
A. E. Tel: 800.984.9503
Tel: 800.767.0329
Tel: 800.829.0146

Louisiana
North/South
A. E. Tel: 800.231.0253
Tel: 800.231.5575

Maine
A. E. Tel: 800.272.9255

Maryland
Baltimore
A. E. Tel: 410.720.3400
W. E. Tel: 800.863.9953

Massachusetts
Boston
A. E. Tel: 978.532.9808
W. E. Tel: 800.444.9953

Michigan
Detroit
A. E. Tel: 313.416.5800
W. E. Tel: 888.318.9953
Grandville
A. E. Tel: 616.531.0345

Minnesota
Minneapolis
A. E. Tel: 612.881.2600
W. E. Tel: 800.860.9953

Mississippi
A. E. Tel: 800.633.2918

Missouri
St. Louis
A. E. Tel: 314.291.5350

Montana
A. E. Tel: 800.526.1741

Nebraska
A. E. Tel: 800.332.4375

Nevada
Las Vegas
A. E. Tel: 800.528.8471
W. E. Tel: 702.765.7117

New Hampshire
A. E. Tel: 800.272.9255

New Jersey
North/South
A. E. Tel: 201.515.1641
Tel: 609.222.6400
Oradell
W. E. Tel: 201.261.3200
Pine Brook
W. E. Tel: 800.862.9953

New Mexico
Albuquerque
A. E. Tel: 505.293.5119

New York
Long Island
A. E. Tel: 516.434.7400
W. E. Tel: 800.861.9953
Rochester
A. E. Tel: 716.475.9130
W. E. Tel: 800.319.9953
Syracuse
A. E. Tel: 315.453.4000

North Carolina
Raleigh
A. E. Tel: 919.872.0712
W. E. Tel: 800.560.9953

North Dakota
A. E. Tel: 800.829.0116

Ohio
Cleveland
A. E. Tel: 216.498.1100
W. E. Tel: 800.763.9953
Dayton
A. E. Tel: 614.888.3313
W. E. Tel: 800.763.9953

Oklahoma
Tulsa
A. E. Tel: 918.459.6000

Oregon
Portland
A. E. Tel: 503.526.6200
W. E. Tel: 800.879.9953

Pennsylvania
Pittsburgh
A. E. Tel: 412.281.4150
Philadelphia
A. E. Tel: 800.526.4812
W. E. Tel: 800.871.9953

Rhode Island
A. E. 800.272.9255

South Carolina
A. E. Tel: 919.872.0712

South Dakota
A. E. Tel: 800.829.0116

Tennessee
East/West
A. E. Tel: 800.241.8182
Tel: 800.633.2918

Texas
Austin
A. E. Tel: 512.219.3700
W. E. Tel: 800.365.9953
Dallas
A. E. Tel: 214.553.4300
W. E. Tel: 800.955.9953
El Paso
A. E. Tel: 800.526.9238
Houston
A. E. Tel: 713.781.6100
W. E. Tel: 800.888.9953
Rio Grande Valley
A. E. Tel: 210.412.2047

Utah
Draper
W. E. Tel: 800.414.4144
Salt Lake City
A. E. Tel: 801.365.3800
W. E. Tel: 800.477.9953

Vermont
A. E. Tel: 800.272.9255

Virginia
A. E. Tel: 800.638.5988

Washington
Seattle
A. E. Tel: 206.882.7000
W. E. Tel: 800.248.9953

West Virginia
A. E. Tel: 800.638.5988

Wisconsin
Milwaukee
A. E. Tel: 414.513.1500
W. E. Tel: 800.867.9953

Wyoming
A. E. Tel: 800.332.9326

Sales Offices and Design Resource Centers

LSI Logic Corporation
Corporate Headquarters
Tel: 408.433.8000
Fax: 408.433.8989

NORTH AMERICA

California
Irvine
◆ Tel: 949.553.5600
Fax: 949.474.8101

Pleasanton Design Center
Tel: 925.730.8800
Fax: 925.730.8700

San Diego
Tel: 619.613.8300
Fax: 619.613.8350

Wireless Design Center
Tel: 619.350.5560
Fax: 619.350.0171

Silicon Valley
◆ Tel: 408.433.8000
Fax: 408.954.3353

Colorado
Boulder
Tel: 303.447.3800
Fax: 303.541.0641

Florida
Boca Raton
Tel: 561.989.3236
Fax: 561.989.3237

Georgia
Roswell
Tel: 770.641.8001
Fax: 770.641.8005

Illinois
Schaumburg
◆ Tel: 847.995.1600
Fax: 847.995.1622

Kentucky
Bowling Green
Tel: 502.793.0010
Fax: 502.793.0040

Maryland
Bethesda
Tel: 301.897.5800
Fax: 301.897.8389

Massachusetts
Waltham
◆ Tel: 781.890.0180
Fax: 781.890.6158

Minnesota
Minneapolis
◆ Tel: 612.921.8300
Fax: 612.921.8399

New Jersey
Edison
◆ Tel: 732.549.4500
Fax: 732.549.4802

New York
Fairport
Tel: 716.218.0020
Fax: 716.218.9010

North Carolina
Raleigh
Tel: 919.785.4520
Fax: 919.783.8909

Oregon
Beaverton
Tel: 503.645.0589
Fax: 503.645.6612

Texas
Austin
Tel: 512.388.7294
Fax: 512.388.4171

Dallas
Tel: 972.503.3205
Fax: 972.503.2258

◆ Tel: 972.244.5000
Fax: 972.244.5001

Houston
Tel: 281.379.7800
Fax: 281.379.7818

Washington
Issaquah
Tel: 425.837.1733
Fax: 425.837.1734

Canada
Ontario
Ottawa
◆ Tel: 613.592.1263
Fax: 613.592.3253

Toronto
Tel: 416.620.7400
Fax: 416.620.5005

Quebec
Montreal
◆ Tel: 514.694.2417
Fax: 514.694.2699

INTERNATIONAL

Australia
New South Wales
Reptechnic Pty Ltd
◆ Tel: 612.9953.9844
Fax: 612.9953.9683

China
Beijing
LSI Logic International Services Inc
Tel: 86.10.6804.2534
Fax: 86.10.6804.2521

France
Paris
LSI Logic S.A.
Immeuble Europa
◆ Tel: 33.1.34.63.13.13
Fax: 33.1.34.63.13.19

Germany
Munich
LSI Logic GmbH
◆ Tel: 49.89.4.58.33.0
Fax: 49.89.4.58.33.108

Stuttgart
Tel: 49.711.13.96.90
Fax: 49.711.86.61.428

Hong Kong
Hong Kong
AVT Industrial Ltd
Tel: 852.2428.0008
Fax: 852.2401.2105

India
Bangalore
Spike Technologies India Private Ltd
◆ Tel: 91.80.664.5530
Fax: 91.80.664.9748

Israel
Ramat Hasharon
LSI Logic
◆ Tel: 972.3.5.480480
Fax: 972.3.5.403747

Netanya
VLSI Development Centre
Tel: 972.9.657190
Fax: 972.9.657194

Italy
Milano
LSI Logic S.P.A.
◆ Tel: 39.039.687371
Fax: 39.039.6057867

Japan
Tokyo
LSI Logic K.K.
◆ Tel: 81.3.5463.7821
Fax: 81.3.5463.7820

Osaka
◆ Tel: 81.6.947.5281
Fax: 81.6.947.5287

Korea
Seoul
LSI Logic Corporation of Korea Ltd
◆ Tel: 82.2.528.3400
Fax: 82.2.528.2250

The Netherlands
Eindhoven
LSI Logic Europe Ltd
Tel: 31.40.265.3580
Fax: 31.40.296.2109

Singapore
Singapore
LSI Logic Pte Ltd
◆ Tel: 65.334.9061
Fax: 65.334.4749

Tel: 65.835.5040
Fax: 65.732.5047

Sweden
Stockholm
LSI Logic AB
◆ Tel: 46.8.444.15.00
Fax: 46.8.750.66.47

Switzerland
Brugg/Biel
LSI Logic Sulzer AG
Tel: 41.32.536363
Fax: 41.32.536367

Taiwan
Taipei
LSI Logic Asia-Pacific
◆ Tel: 886.2.2718.7828
Fax: 886.2.2718.8869

Avnet-Mercuries Corporation, Ltd
Tel: 886.2.2503.1111
Fax: 886.2.2503.1449

Jeilin Technology Corporation, Ltd
Tel: 886.2.2248.4828
Fax: 886.2.2242.4397

Lumax International Corporation, Ltd
Tel: 886.2.2788.3656
Fax: 886.2.2788.3568

Prospect Technology Corporation, Ltd
Tel: 886.2.2721.9533
Fax: 886.2.2773.3756

United Kingdom
Bracknell
LSI Logic Europe Ltd
◆ Tel: 44.1344.426544
Fax: 44.1344.481039

◆ Sales Offices with Design Resource Centers