TOSHIBA

TC35815CF Flow Control 10/100Mbps Ethernet Controller

1 9 9 9

PRELIMINARY

DATA SHEET

Published in April, 1999

This page left blank intentionally

This page left blank intentionally

This specification describes a flow control capable Ethernet controller which operates at either 10-Mbit/s or 100-Mbit/s. In half duplex mode, the controller implements the IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. In full duplex mode, the controller implements IEEE 802.3x MAC Control Layer and the PAUSE Operation for flow control. The controller also supports flow control in half duplex mode, and includes programmable support for additional MAC Control functions. The controller supports direct connection to the 32-bit Peripheral Components Interconnect (PCI) Local Bus, and uses bus master burst transfer mode to efficiently move data to and from system memory. The controller has on-chip memory for buffering, so there is no need for external local buffer memory.

The new 100Mbps implementations of Ethernet increase the capacity of a network ten times, while using existing twisted-pair wiring. By keeping the Media Access Control (MAC) layer and the CSMA/CD protocol unchanged, network administrators can quickly adapt newer and faster implementations. This approach also lowers cost of deployment since it allows re-use of existing applications software. By supporting both 10- and 100Mbps speeds, new products may be deployed into markets which are in transition.

One recent advance in Ethernet is the use of full duplex links and switching hubs to greatly increase the capacity of a local area network. These hubs are more expensive than the traditional shared media and simple repeating hubs. But the increase in throughput is significant for a number of applications. Products which support full duplex are well established in the market place, and the demand for such products is expected to grow.

As part of the evolving support for the full duplex mode of operation, the IEEE 802.3x standards group has approved a new standard for flow control. This standard establishes an optional "MAC Control" sublayer and defines the PAUSE Operation, which is supported by this MAC Control sublayer. The PAUSE Operation provides an industry method of supporting flow control in full duplex Ethernet networks.

Toshiba has designed its TC35815CF Flow Control 10/100Mbps Ethernet Controller for use in highly integrated and cost-effective Ethernet solutions. By supporting direct connection to a PCI bus, the chip can be used with a media driver chip to provide a minimum-parts solution.

The TC35815CF Flow Control 10/100Mbps Ethernet Controller is register compatible with previous Toshiba designs, insuring that investments in system and software development can be leveraged in full duplex systems with flow control.

The controller supports an optional 10Mbps interface. The "7-wire interface" supports mother board applications, where systems developers wish to support older 10 Mb/s PHY’s to ensure interoperability on installed 10 Mb/s networks.

Document Definition

A complete Ethernet circuit is divided into three sections, as shown in Figure 1-1:

1. The system bus interface and Direct Memory Access (DMA) engine.

2. The Media Access Control (MAC) layer.

3. The physical or Medium Dependent Interface (MDI) layer.

The PCI bus interface section contains separate first-in, first-out (FIFO) buffers for transmit and receive, a DMA controller, and a register access module.

The MAC layer consists of transmit and receive blocks, the flow control block, a Content Addressable Memory (CAM) for recognizing addresses, and a number of control, status, and error counter registers. The DMA-Independent Interface (DII) is the interface between the MAC and the DMA section.

The TC35815CF Flow Control 10/100Mbps Ethernet Controller supports the Media Independent Interface (MII). The MII is a standard for a media-independent layer which separates physical-layer issues from the MAC layer. The MII is part of the ISO approved IEEE 802.3 100BASE-T standard for 100Mbps Ethernet.

This document specifies a single chip which implements a PCI bus interface, the DMA engine, and the MAC layer. It is intended as an interface specification and an architectural overview of the device and its operation.

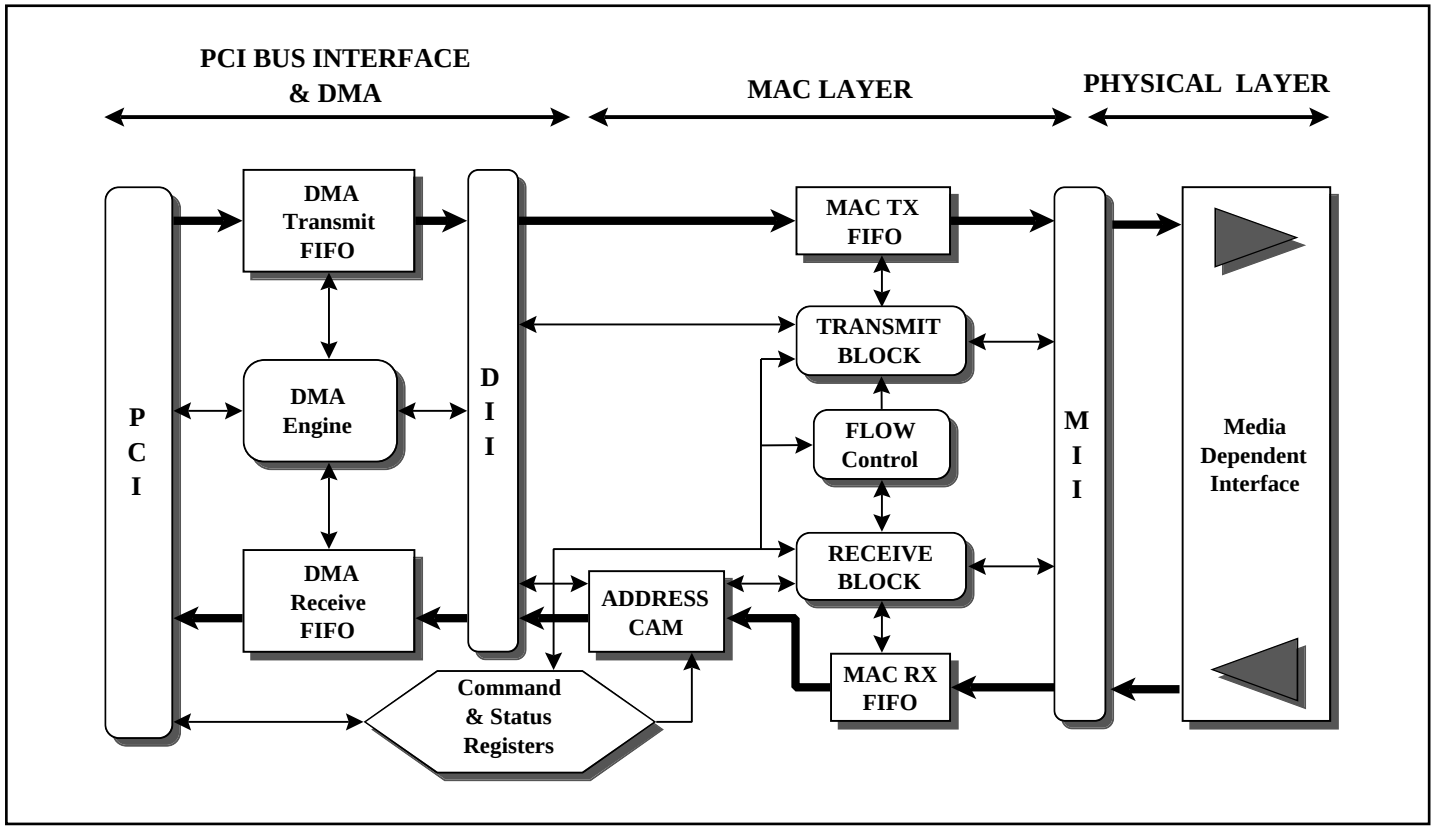


Figure 1-1 Ethernet System Overview

Features and Benefits

Table 1-1 summarizes the features and benefits of the TC35815CF Flow Control 10/100Mbps Ethernet Controller.

Table 1-1 Features and Benefits

Features	Benefits
PCI bus interface.	Direct connection, with no glue logic.
DMA engine using burst mode.	Efficient data transfers.
Large DMA FIFO buffers.	No external buffer memory required.
16-Byte Receive FIFO buffer	Allow DMA Latency during PCI bursts.
80-Byte Transmit FIFO buffer	DMA latency; retransmit after collision without DMA action.
Data alignment logic.	Full data alignment freedom; high bus utilization.
Endian translation.	Operates with either Big-Endian or Little-Endian processors.
Support for old and new media.	Compatible with existing 10Mbps networks.
100/10Mbps operation.	Range of price/performance points. Phased conversion.
Full IEEE 802.3 compatibility.	Compatible with existing applications software.
MII compliant interface.	Can be used with many 100BASE-T physical layers.
Station management signaling	External physical layer configuration and link negotiation.
On-chip CAM	Address recognition for network traffic filtering.
Optional external CAM	Support hub and bridge applications with many addresses.
Optional ROM or EEPROM	Network address and configuration information.
Optional 7-wire Interface	Allow use of old 10Mps physical layer for compatibility
Full duplex mode	Doubles bandwidth.
PAUSE Operation	H/W support for full duplex flow control.
Flexible MAC Control Support	Software can support future MAC Control Operations.
Long packet mode	Specialized environments.
Short packet mode	Specialized environments. Fast testing.
PAD generation	Ease of processing, reduced processing time.
Transmit polling mode	Minimize system overhead to initiate transmission.
Transmit wake-up control	Minimize system latency to transmission initiation.
Receive early-notify control	Minimize system latency to processing received packet.
Power Management	Minimize power consumption connected to quiet network.

PCI System Block Diagram

Figure 1-2 shows a typical block diagram for a computer using the TC35815CF Flow Control 10/100Mbps Ethernet Controller. The implementation may integrate various functions in a slightly different way. For example, the memory controller may be implemented as a single controller, or as separate address and data controllers. Also, the variety of available audio and video compression and other peripherals open up a wide range of possibilities.

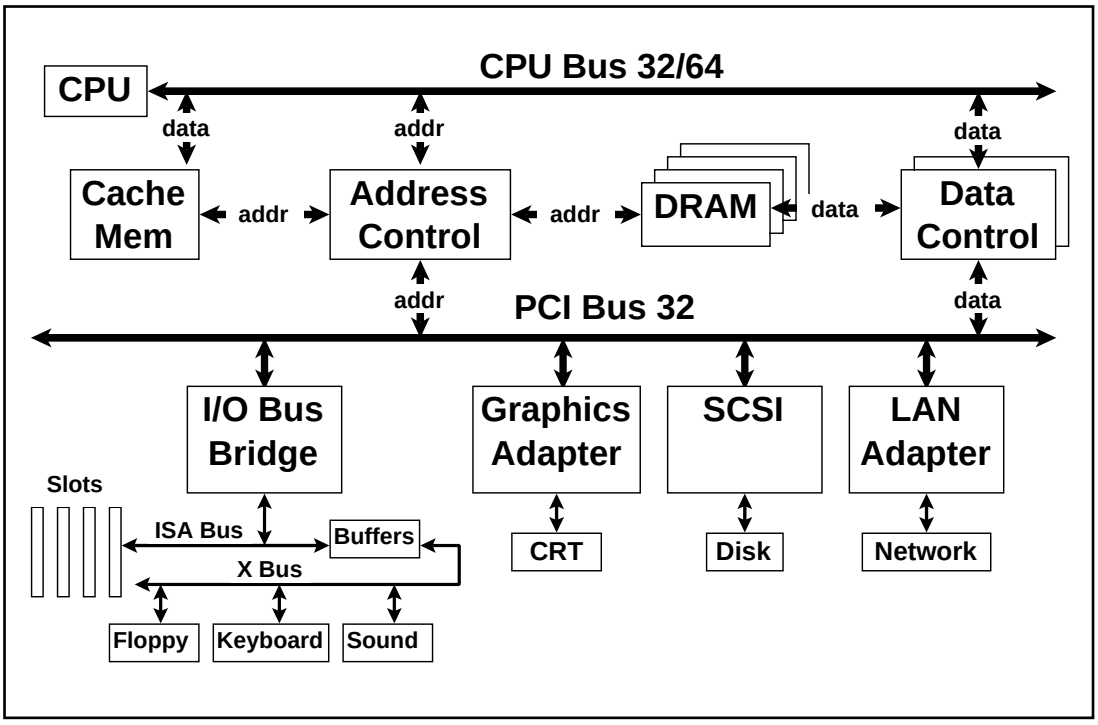


Figure 1-2 PCI System Block Diagram

Controller Application Configurations

Figure 1-3 shows the components that are likely to be used with the TC35815CF Flow Control 10/100Mbps Ethernet Controller. The controller is normally connected to a 100BASE-T medium which can operate at either 100Mbps or 10Mbps. The MII provides transmit and receive clocks for four-bit parallel operation. The clocks operate at 25MHz for 100Mbps operation, or 2.5MHz for 10Mbps operation.

Optionally, the controller can also be connected to 10BASE-T, to provide customer selection of either old or fast Ethernet in the field. The controller supports a 10-MHz clock rate for serial 10Mbps operation, when connected to a 10Mbps Manchester encoder-decoder (endec).

The system designer has additional options: (1) One or more external CAMs can add many Ethernet addresses for the controller to receive. This is useful for applications such as switching hubs, routers, and bridges, where more addresses must be matched with precision than the controller supports with its on-chip CAM. (2) An

inexpensive serial programmable ROM can provide the controller with its Ethernet station address automatically upon power-up. (3) An MII connector can be provided to allow the use of alternate external 100Mbps PHYs.

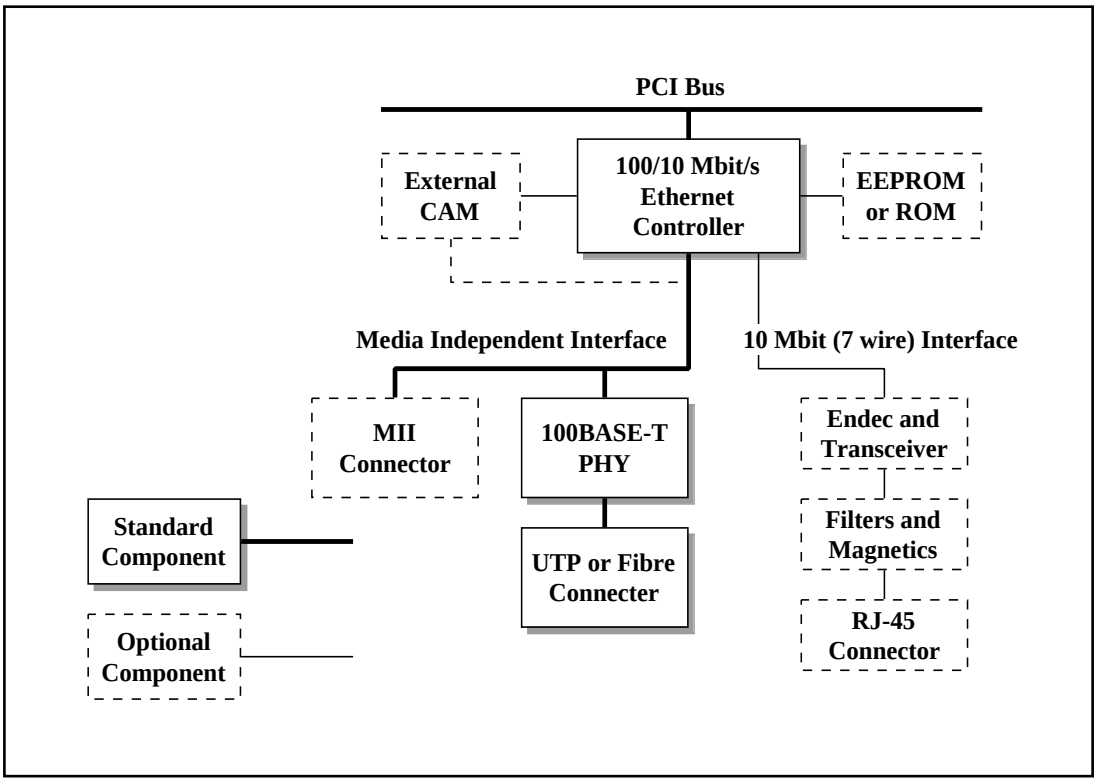


Figure 1-3 Configuration for 100BASE-T Showing Options

Data Structures

The TC35815CF Flow Control 10/100Mbps Ethernet Controller exchanges control information and data via three data structures:

- Frame descriptors
- Buffer descriptors
- Data buffers

Figure 1-4 shows how these data structures are related.

Frame descriptors have a four-byte pointer to the next frame, a field for the system or applications program to use exclusively, a frame status field, a control field for the entire frame, and an array of associated buffer descriptors. Buffer descriptors have a pointer to a data buffer and control field for the buffer. A data buffer is an array of bytes, which can be stored in either little endian or big endian order. For more details on these data structures, see Chapter 5, “Memory Organization.”

The DMA engine supports transfer of data on byte boundaries. Frame descriptors must be aligned on 16 byte boundaries. Buffer descriptors must be aligned on eight-byte boundaries. The DMA engine employs bursts of full four-byte, aligned

transfers whenever possible. Alignment in data buffers does not affect performance very much. The DMA engine avoids doing unaligned or partial word accesses, except at the beginning or end of block transfers.

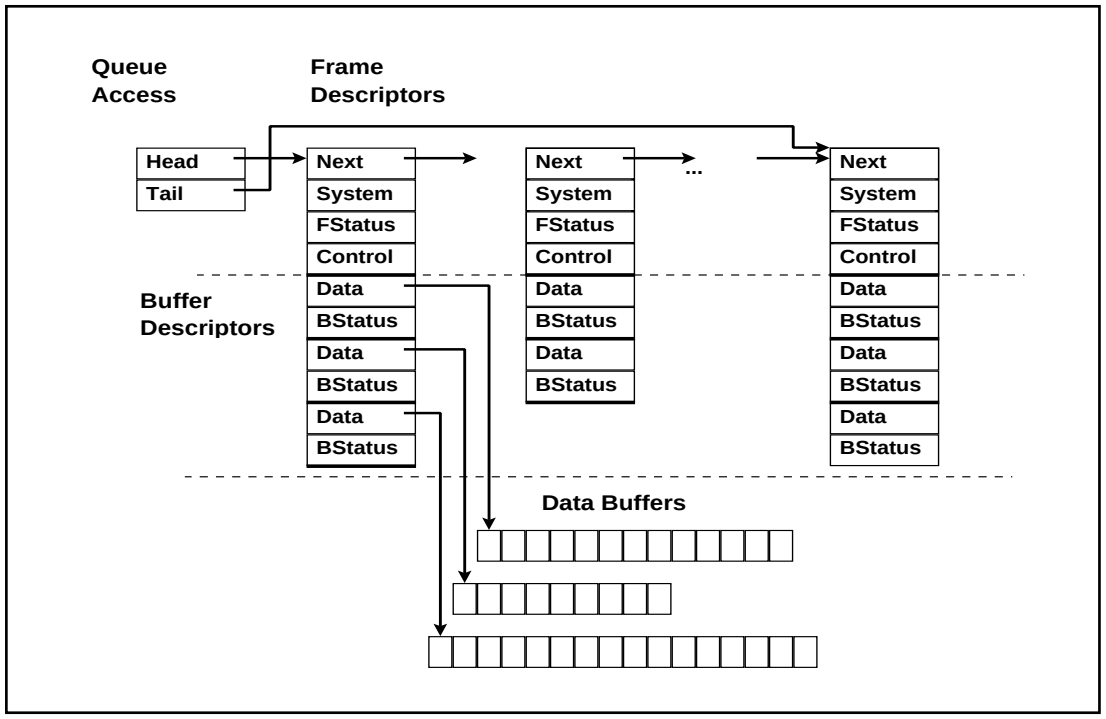


Figure 1-4 Data Structure Overview

System Control Models

The TC35815CF Flow Control 10/100Mbps Ethernet Controller can be programmed for two modes of operation.

- Interrupt on each packet, or group of packets, transmitted or received.
- Poll continuously for new packets to transmit. Post data and descriptors, but do not interrupt for packets received.

An interrupt for each packet or group of packets is the traditional way that Ethernet software is controlled. This is appropriate for occasional network traffic, such as electronic mail.

However, as processors increase in speed and complexity, the time to service an interrupt does not improve correspondingly. In fact, many modern processors have large amounts of state information and sophisticated caching schemes, which can slow the relative speed of interrupt processing even further. Combined with a faster network, where packets may arrive more frequently, the overhead for servicing network interrupts can become a significant burden to the system. Also, some applications, such as interactive conferences, require a much higher level of network traffic. These factors lead to the design of more efficient network control schemes.

By enabling or disabling interrupts for selected frame descriptors, the system can arrange for the controller to process multiple packets between interrupts. This

reduces the overhead in servicing interrupts, and improves performance by caching code when the system processes several packets at once.

It is even possible to set up the controller so that it generates no interrupts at all unless serious errors occur. This requires enabling some additional controls, to ensure that the starting and stopping of traffic, and polling during idle times, is handled efficiently. For more details, see Chapter 7, “DMA Operation.”

Power Management

The TC35815CF Flow Control 10/100Mbps Ethernet Controller provides for dynamic power management. By reducing the clock rate to idle circuits, the controller consumes significantly less power, while maintaining full network functionality.

Given the current emphasis on the “green desktop”, or PCs which meet governmental guidelines for energy efficiency, power management is becoming an issue that affects a broader market segment than just lap-tops and palm-tops. The TC35815CF Flow Control 10/100Mbps Ethernet Controller is designed to provide a fully-functional, yet power efficient solution for networking.

Reference Documents

See the following documents for additional technical information:

PCI Local Bus Specification, Revision 2.1, June 1, 1995. PCI Special Interest Group, Hilsboro, Oregon, (800) 433-5177.

PCI Local Bus Specification, Production Version, Revision 2.0, April 30, 1993. PCI Special Interest Group, Hilsboro, Oregon, (800) 433-5177.

International Standard ISO/IEC 8802-3, ANSI/IEEE Std 802.3. “*Information technology—Telecomunications and Information exchange between systems—Local and metropolitan area networks—Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*,” Fifth Edition 1996-07-29.

IEEE Std 802.u-1995*(Supplement to International Standard ISO/IEC 8802-3: 1993 [ANSI/IEEE Std 802.3*, 1993 Edition]). IEEE Standards for Local and Metroploitan Area Networks: Supplement to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. "Media Access Control (MAC) Parameters, Physical Layer, Medium Attachment Units and Repeater for 100 Mb/s Operation, Type 100Base-T. (Clauses 21-30)". Approved June 14, 1995.

PCI Hardware and Software, by Edward Solari and George Willse, Second Edition, 1995. Annabooks, San Diego, CA, (619) 673-0870.

PCI System Architecture, Third Edition, by Tom Shanley and Don Anderson, February 1995. MindShare, Inc., published by Addison-Wesley.

IEEE Std. 802.3x - 1997 Supplement to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications:

"Specification for 802.3 Full Duplex Operation".

Project 802 Local & Metropolitan Area Networks, Draft Guide to ANSI/IEEE Std 802.3 (CSMA/CD Access Method & Physical Layer Specifications), SYSTAG Network Guide, Draft 5.2, January 23, 1994.

The Ethernet. A Local Area Network. Data Link Layer and Physical Layer Specifications. Digital Equipment Corporation, Intel Corporation, and Xerox Corporation. Version 2.0. (*"The Blue Book"*) November, 1982.

Signal and Power Supply Information

Figure 2-1 shows the 94 external signals for the TC35815CF Flow Control 10/100Mbps Ethernet Controller, divided into functional groups. This chapter groups the signal definitions by functional area, giving each signal's symbolic name, full name, direction, clock domain, and brief definition. The groups are:

- 37 for PCI address and data
- 13 for PCI control, arbitration, error reporting, interrupt, and system
- 18 for MII Transmit and Receive
- 9 for internal scan
- 4 for an optional external EEPROM/ROM
- 9 for an optional 10Mbps Manchester encoder-decoder (endec)
- 2 for an optional external CAM interface
- 5 for internal test

Table 2-1 Power Supply Information

Signal	Pin #	Function
VSS	1, 9, 19, 25, 31, 40, 44, 49, 52, 57, 64, 67, 73, 78, 81, 85, 88, 91, 96, 102, 110, 116, 118, 123, 127, 136	Ground of power supply (0V)
VDD	6, 16, 18, 28, 37, 48, 55, 60, 68, 80, 82, 90, 94, 104, 109, 117, 126, 129, 140	3.3V power supply

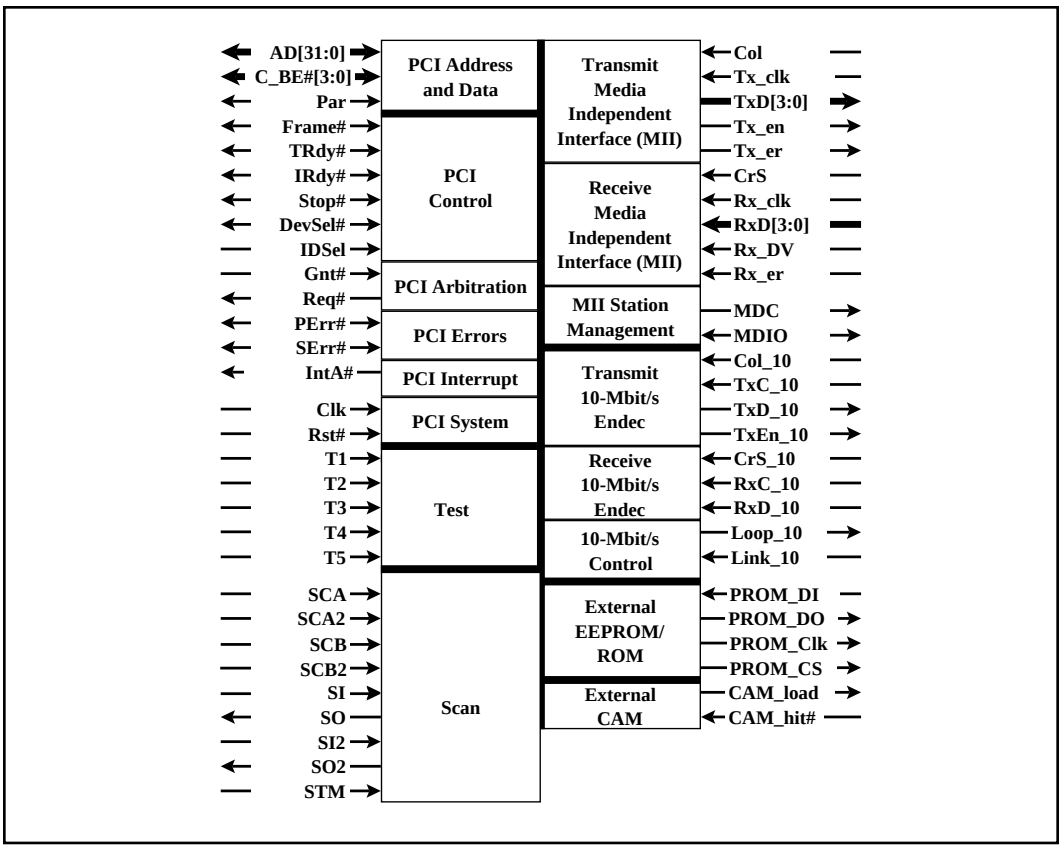


Figure 2-1 External Signals

Peripheral Component Interconnect (PCI) Signals

Table 2-2 shows the 50 Peripheral Component Interconnect (PCI) signals for the TC35815CF Flow Control 10/100Mbps Ethernet Controller. These include the 49 signals required for a PCI master, plus the Interrupt A signal. The part drives sustained tri-state signals high for one clock before returning them to the high-impedance state. Signals whose symbols end in "#" are active-low signals. All signals are referenced to the rising edge of Clk, except SErr#, Rst# and IntA#. The SErr# signal is asserted synchronous, and deasserted asynchronous to the Clk.

Table 2-2 Peripheral Component Interconnect (PCI) Signals

Function	Symbol	Name	Type	Master	Target	Clock domain
Address and data	AD[31:0]	Address and data bus	Tri-state	In/Out	In/Out	PCI clock
	C_BE#[3:0]	Command and byte enable bus	Tri-state	Out	In	PCI clock
	Par	Parity	Tri-state	In/Out	In/Out	PCI clock
Interface control	Frame#	Cycle frame	Sustained tri-state	Out	In	PCI clock
	TRdy#	Target ready	Sustained tri-state	In	Out	PCI clock
	IRdy#	Initiator ready	Sustained tri-state	Out	In	PCI clock
	Stop#	Stop	Sustained tri-state	In	Out	PCI clock
	DevSel#	Device select	Sustained tri-state	In	Out	PCI clock
	IDSel	Initialization device select	Input	--	In	PCI clock
Arbitration	Gnt#	Grant	Input[a]	In	--	PCI clock
	Req#	Request	Tri-state	Out	--	PCI clock
Error reporting	PErr#	Parity error	Sustained tri-state	In/Out	In/Out	PCI clock
	SErr#	System error	Open drain	In[b]	Out[c]	PCI clock/Asynch
Interrupt	IntA#	Interrupt A	Open drain	Out	Out	Asynchronous
System	Clk	PCI clock	Input	In	In	PCI clock
	Rst#	Reset	Input	In	In	Asynchronous

a. Gnt# is a point-to-point signal. The PCI Specification specifies the use of a tri-state driver.
b. In master mode, the controller does not signal SErr#, but monitors it as an input.
c. In slave mode, the controller signals SErr# for address parity errors.

PCI Address and Data Signals

AD[31:0]

Address and data bus
Carries address information during the address phase, and data during the data phase.

C_BE#[3:0]

Command and byte enable

During the address phase, C_BE#[3:0] defines the type of transaction, and during the data phase, it indicates the validity of bytes carried on AD[31:0]. Table 2-3 shows the correspondence between the byte-enable signals and bytes on the address and data bus during the data phase.

Table 2-3 Correspondence Between C_BE#[3:0] and AD[31:0]

Address and data bus AD[31:0]																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte 3 (MSB)								Byte 2								Byte 1								Byte 0 (LSB)							
C_BE#[3]								C_BE#[2]								C_BE#[1]								C_BE#[0]							

Table 2-4 Big Endian Byte Order

Address and data bus AD[31:0]																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte 0 (MSB)								Byte 1								Byte 2								Byte 3 (LSB)							
C_BE#[3]								C_BE#[2]								C_BE#[1]								C_BE#[0]							

Table 2-4 shows the byte ordering of data when Big Endian is enabled.

Par

Parity

Provides even parity for the information carried on AD[31:0] and C_BE#[3:0], by causing AD[31:0], C_BE#[3:0], and Par to contain an even number of bits equal to one. The Par signal is active during the clock period after AD and C_BE# contain valid data.

PCI Control Signals

Frame#

Cycle frame

Driven by the current master to signal how long an access lasts. In a bus transaction terminated by the master, the master asserts Frame# beginning with the first or address cycle, and holding it until the next to last cycle, when it deasserts it. (See the STOP# signal for a description of target abort.)

TRdy#

Target ready

During a read, the target asserts TRdy# to indicate that it is driving valid data on AD[31:0]. During a write, the target asserts TRdy# to indicate it is ready to receive data over AD[31:0]. A data transfer occurs in a cycle when both TRdy# and IRdy# are asserted.

IRdy#

Initiator ready

During a read, the initiator asserts IRdy# to indicate that it is ready to receive data on AD[31:0]. During a write, the initiator asserts IRdy# to indicate it is driving valid data over AD[31:0]. A data transfer occurs in a cycle when both TRdy# and IRdy# are asserted.

Stop#

Stop#

The target asserts Stop# to request that the master terminate or abort the current transaction.

DevSel#

Device select

Driven by the target to confirm that it has decoded the address as referring to itself.

IDSel

Initialization device select

Equivalent to chip select, IDSel is a fully-decoded addressing mechanism for configuration read and write transactions.

PCI Bus Arbitration Signals

Gnt#

Grant

A point-to-point signal from the arbiter to an agent, signalling to the agent that the arbiter has granted it bus ownership.

Req#

Request

A point-to-point signal from an agent to the arbiter, signalling to the arbiter that

the agent desires bus ownership. In normal operation, Req#, is an output, but during reset it enters a high-impedance state.

PCI Error Signals

PErr#

Parity error

Kept at a high-impedance state, unless an agent receives non-Special Cycle data with a parity error, in which case it asserts PErr# two clocks later, and then, as with all sustained tri-state signals (unlike SErr#), drives it high for one clock before returning to the high-impedance state.

SErr#

System error

The SErr# signal is used to signal parity errors during an address cycle of a bus transaction, and all errors other than parity errors.

PCI Interrupt Signal

IntA#

Interrupt A

Requests an interrupt.

PCI System Signals

Clk

Clock

All inputs are sampled on rising edge of clock. The controller supports 100 Mbps Ethernet when operating at 16-33MHz frequency operation. May be held low at any time, to conserve power. May only be stopped in a low state.

Rst#

Reset

Causes all output signals to enter a high-impedance state, and clears all registers. Does not affect on-chip RAM or FIFOs. Upon deassertion, the software drivers are responsible to check for the presence of a serial ROM, and if present, to read in the station address, and other configuration parameters.

Media Independent Interface (MII) Signals

The MII is the interface between the TC35815CF Flow Control 10/100Mbps Ethernet Controller and the Physical Layer (PHY).

Transmit MII Signals

Table 2-5 shows the MII signals supported by the TC35815CF Flow Control 10/100Mbps Ethernet Controller for transmitting packets. For a detailed description of these signals, see the MII sections of the 802.3u documents listed in the “Reference Documents” section.

Table 2-5 Transmit MII Signals

Symbol	Name	Direction	Clock domain
Col	Collision	Input	–
Tx_clk	Transmit clock	Input	–
TxD[3:0]	Transmit data	Output	Tx_clk
Tx_en	Transmit enable	Output	Tx_clk
Tx_er	Transmit coding error	Output	Tx_clk

Col

Collision

Asserted asynchronously with minimum delay from the start of a collision on the medium.

Tx_clk

Transmit clock

TxD[3:0] and Tx_en are driven off the rising edge of the Tx_clk by the controller, and sampled by the PHY on the rising edge of the Tx_clk.

TxD[3:0]

Transmit data

Transmit data is aligned on nibble boundaries. TxD[0] corresponds to the first bit to transmit on the physical medium and is the LSB of the first byte, followed by the fifth bit of that byte during the next clock.

Tx_en

Transmit enable

Tx_en provides precise framing for the data carried on TxD[3:0]. It is active during the clock periods that TxD[3:0] contains valid data to be transmitted, from preamble through CRC.

Tx_er

Transmit coding error

Tx_er is driven synchronously to Tx_clk and is sampled continuously by the physical layer entity (PHY). If asserted for one or more Tx_clk periods, it causes the PHY to emit one or more symbols which are not part of the valid data or delimiter set somewhere in the frame being transmitted.

Receive MII Signals

Table 2-6 shows the MII signals supported by the TC35815CF Flow Control 10/100Mbps Ethernet Controller for receiving packets. For a detailed description of these signals, see the MII sections of the 802.3u documents listed in the “Reference Documents” section.

Table 2-6 Receive MII Signals

Symbol	Name	Direction	Clock domain
CrS	Carrier sense	Input	–
Rx_clk	Receive clock	Input	–
RxD[3:0]	Receive data	Input	Rx_clk
Rx_DV	Receive data valid	Input	Rx_clk
Rx_er	Receive error	Input	Rx_clk

CrS

Carrier sense

Asserted asynchronously with minimum delay from the detection of a non-idle medium.

Rx_clk

Receive clock

Rx_clk is a continuous clock. In four-bit mode, its frequency is 25MHz for 100Mbps operation, and 2.5 MHz for 10Mbps. RxD[3:0], Rx_DV, and Rx_er are driven by the PHY off the falling edge of Rx_clk, and sampled on the rising edge of Rx_clk.

RxD[3:0]

Receive data

RxD is aligned on nibble boundaries. RxD[0] corresponds to the first bit received on the physical medium which is the LSB of the byte in one clock period and the fifth bit of that byte in the next clock.

Rx_DV

Receive data valid

PHY asserts Rx_DV synchronously and holds it active during the clock periods that RxD[3:0] contains valid received data. PHY asserts Rx_DV no later than the clock period when it places the first nibble of the start frame delimiter (SFD) on RxD[3:0]. If PHY asserts Rx_DV prior to the first nibble of the SFD, then RxD[3:0] carries valid preamble symbols.

Rx_er

Receive error

PHY asserts Rx_er synchronously whenever it detects a physical medium error, e.g., a coding violation. PHY asserts Rx_er only when it asserts Rx_DV.

MII Station Management Signals

Table 2-7 shows the two MII station management signals. Use of these signals for configuring a PHY or negotiating a link protocol is optional.

Table 2-7 MII Station Management Signals

Symbol	Name	Direction
MDC	Management Data Clock	Output
MDIO	Management Data Input/Output	Tri-state

MDC

Management Data Clock

Timing reference for transfer of information on the MDIO signal. With the PCI clock at 33MHz, the MDC clock has a maximum clock frequency of 33/14 = 2.36MHz. The minimum clock period is 424 ns.

MDIO

Management Data I/O

Transfers data to or from a PHY attached to the MII interface. Controller can initiate a sequence to determine whether a PHY is attached. Output/input is tri-state and a pull down resistor is provided in the pad.

External 10Mbps Endec Signals

These signals support connection to a optional 10BASE-TPHY. This mode of operation is distinct from the 10Mbps operation mode of the MII. The external endec uses a 1-bit serial signal, while the MII supports a 4-bit parallel interface.

Transmit 10Mbps Signals

Table 2-8 shows the four signals which support transmission via an external 10Mbps Manchester encoder-decoder (endec).

Table 2-8 Transmit 10Mbps Endec Interface

Symbol	Name	Direction
Col_10	Collision detect on 10Mbps endec	Input
TxC_10	Transmit clock on 10Mbps endec	Input
TxD_10	Transmit data on 10Mbps endec	Output
TxEn_10	Transmit enable on 10Mbps endec	Output

Col_10

Collision detect on 10Mbps endec

Asserted when a 10Mbps PHY detects a collision. Ignored if 10Mbps PHY is not enabled by the master control register.

TxC_10

Transmit clock on 10Mbps endec

Clock from 10Mbps PHY to transfer data. Ignored if 10Mbps PHY is not enabled by the master control register.

TxD_10

Transmit data on 10Mbps endec

Data line for transmitting to the 10Mbps PHY. Stays low if 10Mbps PHY is not enabled by the master control register.

TxEn_10

Transmit enable on 10Mbps endec

Asserted by the controller when it is ready to transfer data. Stays low if 10Mbps PHY is not enabled by the master control register.

Receive 10Mbps Signals

Table 2-9 shows the three signals which support reception via an external 10Mbps endec.

Table 2-9 Receive 10Mbps Endec Interface

Symbol	Name	Direction
CrS_10	Carrier sense on 10Mbps endec	Input
RxC_10	Receive clock on 10Mbps endec	Input
RxD_10	Receive data on 10Mbps endec	Input

CrS_10

Carrier sense on 10Mbps endec

Asserted when a 10Mbps PHY has data to transfer. Ignored if 10Mbps PHY is not enabled by the master control register.

RxC_10

Receive clock 10Mbps endec

Clock from 10Mbps PHY to receive data. Ignored if 10Mbps PHY is not enabled by the master control register.

RxD_10

Receive data 10Mbps endec

Data line for receiving from the 10Mbps PHY. Ignored if 10Mbps PHY is not enabled by the master control register.

10Mbps Control Signals

Table 2-10 shows the two signals provided to control 10Mbps PHY. Use of these signals is optional.

Table 2-10 10Mbps Control Signals

Symbol	Name	Direction
Loop_10	Loop back at 10Mbps endec	Output
Link_10	Link status of 10Mbps endec	Input

Loop_10

Loop back at 10Mbps endec

Output signal, driven by bit 7 of MAC of Control Register.

Link_10

Link status of 10Mbps endec

Used to convey link status of the 10Mbps endec. Stored in bit 15 of MAC of Control Register. Ignored if 10Mbps PHY is not enabled by the master control register.

External EEPROM or ROM Interface

Table 2-11 shows the four signals which control an external EEPROM or ROM. Use of these signals is optional.

Table 2-11 External EEPROM or ROM Interface

Symbol	Name	Direction
PROM_DI	EEPROM/ ROM data input	Input
PROM_DO	EEPROM/ ROM data output	Output
PROM_Clk	EEPROM/ ROM clock	Output
PROM_CS	EEPROM/ ROM chip select	Output

PROM_DI

EEPROM/ROM data input

Data line for transmitting from external EEPROM/ROM to the controller. Must be high with no EEPROM present. An internal resistor pull-up is a solution.

PROM_DO

EEPROM/ROM data output

Transfers data from the controller to an external EEPROM/ROM.

PROM_Clk

EEPROM/ROM clock

Clock for transmitting to and from an external EEPROM/ROM. With the PCI clock at 33MHz, the maximum clock frequency for PROM-Clk is 33/34 = 0.97MHz. This is compatible with the slowest commercial parts, which specify a maximum frequency of 1MHz.

PROM_CS

EEPROM/ROM chip select

Used to frame transmissions to and from an external EEPROM/ROM.

External CAM Interface Signals

Table 2-12 shows the two external CAM interface signals. Use of these signals is optional.

Table 2-12 External CAM Interface Signals

Symbol	Name	Direction
CAM_load	External CAM address load	Output
CAM_hit#	External CAM hit	Input

CAM_load

External CAM address load

Signals the external CAM to begin loading the destination address from RxD[3:0]. This signal is optional even with an external CAM, as external circuitry can derive it from MII signals, just as the MAC does.

CAM_hit#

External CAM hit

Notification from the external CAM that it has recognized this packet's destination address. Held inactive by a pull-up resistor if the external CAM is not present.

Internal Scan Interface Signals

Table 2-13 lists nine scan interface signals. This group of signals is used to implement scan test vectors for automatic testing.

Table 2-13 Internal Scan Interface Signals

Symbol	Name	Direction
SCA	Scan Clock A	Input
SCA2	Scan Clock A-2	Input
SCB	Scan Clock B	Input
SCB2	Scan Clock B-2	Input
SI	Scan Data In	Input
SO	Scan Data Out	Tri-state
SI2	Scan Data In-2	Input
S02	Scan Data Out-2	Tri-state
STM	Scan Test Mode	Input

The TC35815CF Flow Control 10/100Mbps Ethernet Controller requires two scan chains. The Scan Test Mode signal is required to be asserted during scan testing, in order to hold some internal logic elements in a testable state. These include elements related to clock muxing, power management, software reset, and memory blocks.

SCA

Scan Clock A-2
The Scan Clock A for the second internal scan chain.

SCB

Scan Clock B
The Scan Clock B for the first internal scan chain.

SCB2

Scan Clock B-2
The Scan Clock B for the second internal scan chain.

SI

Scan Data Input
The Scan Data Input is the serial input for data shifted into the first internal scan chain.

SO

Scan Data Output
The Scan Data Output is the serial output for data shifted from the first internal scan chain.

SI2

Scan Data Input-2
The Scan Data Input is the serial input for data shifted into the second internal scan chain.

SO2

Scan Data Output-2
The Scan Data Output is the serial output for data shifted from the second internal scan chain.

STM

Scan Test Mode
The Scan Test Mode is asserted during scan testing, and deasserted for normal operation. It is required to place the chip in a testable state. This signal is deasserted when mounted in a user device. A pull-down resistor should be provided internally, to make this the default.

This page left blank intentionally

Overview of Functional Blocks

Figure 3-1 gave an overview of an Ethernet circuit. The functional blocks of the TC35815CF Flow Control 10/100Mbps Controller are shown in more detail in Figures 3-1 and 3-2. Figure 3-1 shows the DMA functional blocks, while Figure 3-2 shows the MAC functional blocks.

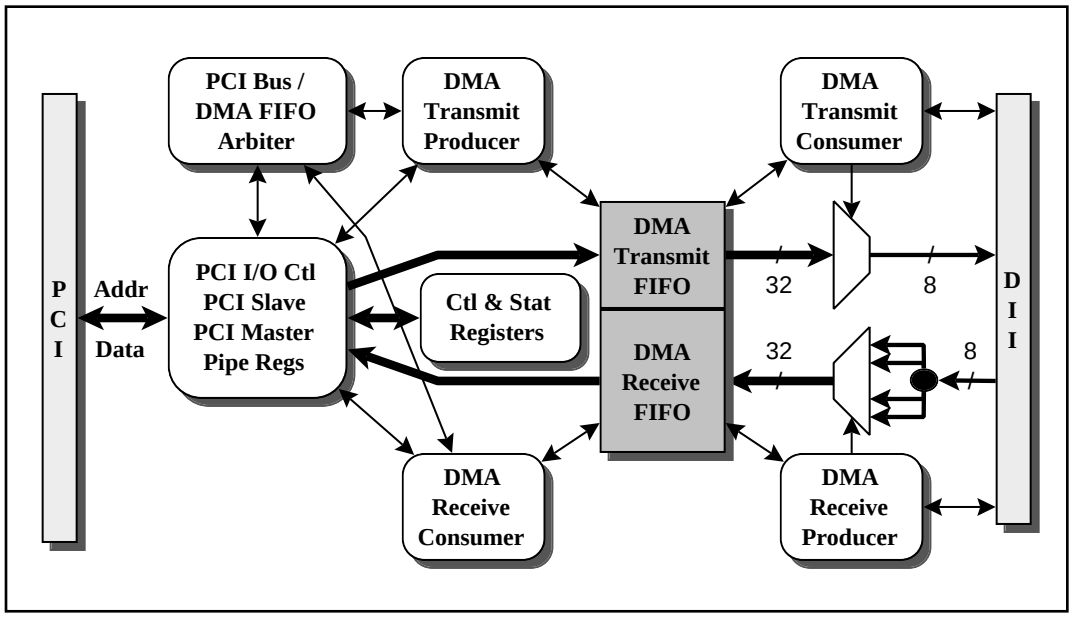


Figure 3-1 DMA Functional Blocks

Both diagrams, refer to the DMA Independent Interface (DII). This is an internal interface with a simple handshaking protocol, which interfaces the two, small, byte-wide FIFOs in the MAC with wider and deeper FIFOs in the DMA block.

PCI and DMA Overview

The Peripheral Component Interconnect (PCI) bus is an industry-standard bus for high performance PCs and workstations. In addition, some network hub designs use the PCI bus for internal data paths, or back plane connections. Figure 3-1 shows the major functional blocks in the DMA for configuring and controlling the PCI bus, and for transferring data to and from the DMA FIFO's:

- PCI I/O Control block, for generating and recognizing PCI Control signals.
- PCI Slave block, for recognizing and controlling transactions where the Ethernet controller is the target device.
- PCI Master block, for controlling transactions where the Ethernet controller is the initiator device.

- PCI Pipe registers, to provide buffering, so DMA engine can sustain back-to-back (or 1-1-1-1...) performance during long burst operations.
- Control and Status Registers, for PCI configuration and MAC and DMA control.
- An Arbiter block, for allocating access to the PCI Bus and DMA memory.
- Producer and Consumer control blocks for Transmit and Receive data entering and leaving the DMA FIFO's.

The PCI Configuration registers and DMA registers provide configuration and control information for the Ethernet Controller as a PCI I/O device. They also provide access to data structures stored in main memory, and access to the control and status information in the MAC.

There are two FIFO buffers controlled by the DMA engine. The DMA Transmit FIFO holds data and status information for packets being transmitted. The DMA Receive FIFO holds data and status information for packets being received. The PCI arbiter decides which of the consumer and producer state machines has highest priority for accessing the FIFO buffers. The priority is dynamic. During burst transfers, the state machines controlling the PCI bus are given priority, if possible. But if a receive FIFO is close to becoming full, or a transmit FIFO is close to becoming empty during transmission, that FIFO will receive priority. Otherwise, the arbiter will provide round-robin fair service.

The DMA Controller blocks provide logic for controlling bus master read and write operations across the PCI bus. They include:

- Burst size control, to optimize PCI and system performance.
- Transmit threshold control, to match transmission latency to PCI bus latency.
- Big Endian byte swapping, to support data transfers to Big Endian processors.
- Buffer spanning and packing controls.
- Polling controls, to optionally poll for packets to transmit.
- Wake-up controls, to start transmission as soon as data is ready.
- Early notification, to allow processing of incoming data to begin before data end.
- Interrupt enable controls, to adjust controller behavior to protocol requirements.

MAC Overview

In Figure 3-2 we see that the MAC consists of a transmit block, a receive block, a set of control and status registers, and two serial controllers, one for accessing the MII station management interface, and one for the external EEPROM/ROM. The MAC also has a loopback circuit, optional support for a 10Mbps interface to a Manchester encode/decoder, and support for an optional external CAM circuit.

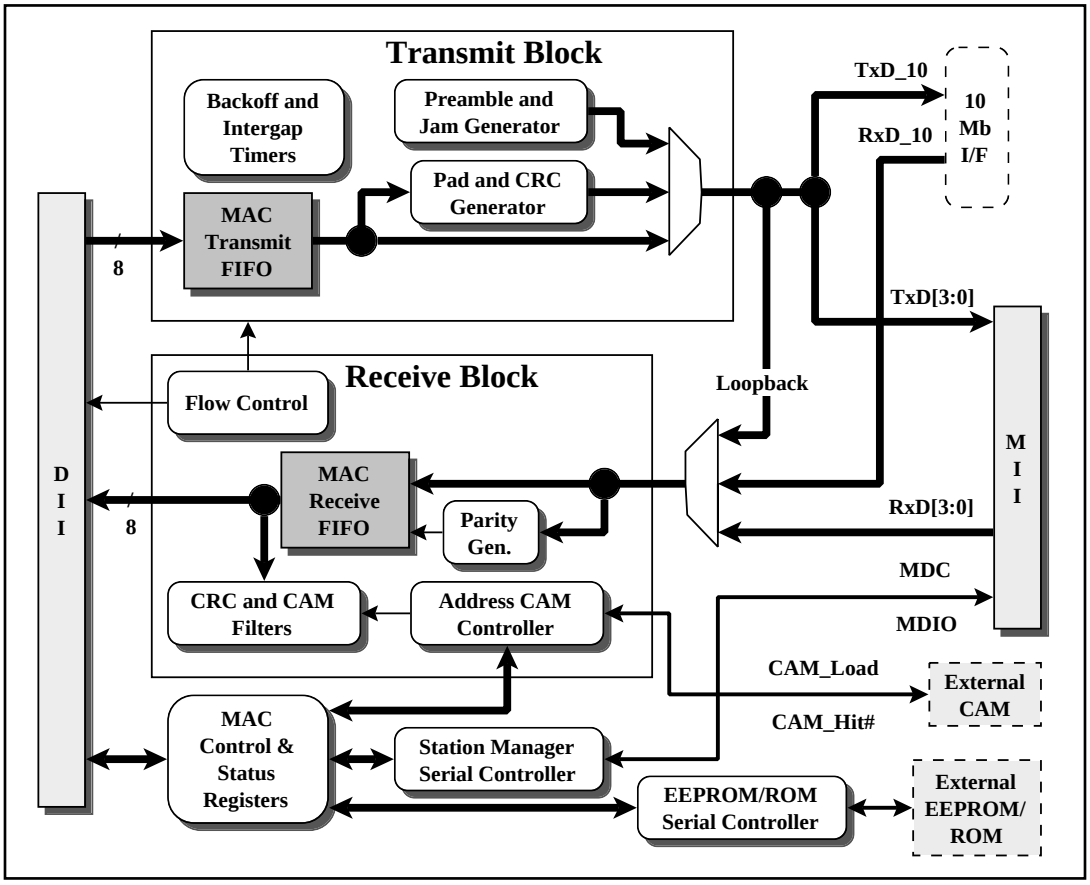


Figure 3-2 MAC Functional Blocks

The Media Independent Interface (MII) is the interface between a 100 BASE-T compatible physical layer and the transmit and receive blocks, and is part of the IEEE 802.3 standard.

The transmit block buffers the outgoing data in the MAC transmit FIFO, encapsulates it, and passes it on to the MII and the 10Mbps interface. The transmit block has circuits for generating preamble and jam bytes, pad bytes, and the CRC value. It also has logic to check parity, a timer for the backoff delay after a collision, and a timer for the inter-packet gap after a transmission.

The receive block decapsulates the received packet from the MII and stores it in the MAC receive FIFO. The receive block has circuits for checking the CRC value, generating parity to protect data in the FIFO, and checking packet lengths. The receive block also has a Content Addressable Memory (CAM) block which provides for acceptance or rejection of a packet based on its destination address.

The registers control programmable options, and specify which conditions interrupt the system. The status registers hold information for error handling software, and the error counters accumulate statistical information for network management software.

The loopback circuit provides for MAC-layer testing in isolation from the MII and physical layer.

The MAC blocks provide controls for network operation, including:

- Controls to enable and disable transmit and receive circuit, including requests to halt at end of current packet.

- Interrupt enable and disable controls for individual conditions.
- Address recognition controls, for up to 21 individual addresses.
- Counters and status bits for collecting network management data.
- Loopback and other controls, to aid in diagnosing problems.
- Controls for reading, writing, and erasing values in an EEPROM or ROM chip.
- PAUSE operation enable controls, to enable pausing of Transmitter on receipt of MAC Control packets specifying a PAUSE operation.
- MAC Control packets transmit controls, to enable generation of PAUSE and other MAC Control packets, even when transmitter is paused.
- MAC Control packets pass through controls, to enable S/W to process other forms of MAC Control packets.

Media Independent Interface (MII)

Both transmit and receive blocks operate using the MII, which was developed by the IEEE 802.3 Task Force on 100Mbps Ethernet to support the following goals:

1. Physical media independence.
2. Multi-vendor point of interoperability.
3. Data and data delimiters are synchronous to clock references.
4. Provides independent four-bit wide transmit and receive data paths.
5. Capable of supporting both 100 and 10Mbps rates.
6. Compatible with common digital CMOS ASIC processes.
7. Primarily a chip-to-chip interface with TTL signal levels.
8. Provides a simple management interface for speed detection, link status, etc.
9. Support connection of MAC layer end Physical (PHY) layer devices.
10. Capable of driving a limited length of shielded cable via a standard connector.

The physical layer or PHY encodes outgoing data and decodes incoming data. The manner of encoding (Manchester for 10Mbps operation, 4B/5B for 100BASE-X, or 8B/6T for 100BASE-T) does not affect the MII. The MII expects raw data for transmission, starting with the preamble and ending with the CRC. The MAC layer can also generate jam and pad bytes. The station manager block, described in this chapter on page 30, “MII Station Manager,” provides an interface for speed detection, link status, etc.

Chapter 2, “External Signals,” defines the signals which make up the MII.

Chapter 8, “Timing,” gives detailed timing information for the MII.

Peripheral Component Interconnect (PCI) Bus

The Peripheral Component Interconnect (PCI) bus is the latest in a series of local bus standards for PC- and workstation-class machines. The primary reasons for supporting the PCI bus are:

- Industry standard – both PC and workstation companies are supporting it.
- High speed – fast enough to support 100Mbps Ethernet.
- Cost effective – chip sets in high volume will ensure low costs.
- Bus master mode – relieves the system of significant processing load.

The PCI bus features 32-bit addresses and a 32-bit data path, multiplexed over the same pins. For future growth, a 64-bit extension of the bus is also defined, but is not supported by this controller. In burst-mode transfers, the PCI bus can, in theory, support 132MBytes transfer rates (the 64-bit version supports 264MBytes) with a 33MHz system clock. The TC35815CF Flow Control 10/100Mbps Ethernet Controller works with both current and future higher performance bus controller chips.

Chapter 2, “External Signals,” defines the signals which make up the PCI bus.

Chapter 8, “Timing,” gives detailed timing information for the PCI bus.

DMA Functional Blocks

PCI Arbiter

The PCI arbiter accepts as input signals status information from the DMA transmit and receive controllers. The PCI arbiter controls switching between transmit, receive, and notification activity on the PCI bus using PCI in master mode access. The PCI arbiter also allows slave mode access to read and write control and status registers.

DMA Transmit Controller

The DMA transmit controller consists of two state machines: a producer and a consumer. The DMA transmit producer reads the transmit queue frame descriptor, and controls the transfer of data from the transmit queue to the transmit data FIFO. It also controls the writing of transmit status information, which gives MAC status information about transmitted packets, after transmission is complete. The DMA transmit consumer controls the MAC transmit engine and the movement of data from the DMA transmit FIFO to the smaller MAC transmit FIFO.

DMA Receive Controller

The DMA receive controller consists of two state machines: a producer and a consumer. The DMA receive producer controls the movement of data from the MAC to the DMA receive FIFO. The DMA receive consumer allocates buffers from the free buffer list, writes new frame descriptors and related buffer descriptors in the free descriptor area, and controls the transfer of bytes from the DMA receive FIFO into system memory via the PCI bus.

MAC Functional Blocks

MAC Transmit Block

The transmit block is responsible for transmitting data. It complies with the IEEE 802.3 Standard for Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. It also supports a full-duplex mode of operation, which allows simultaneous transmission while receiving.

The transmit block consists of the following:

1. Transmit FIFO, FIFO controller, and counters
2. Preamble and jam generator
3. Pad byte and CRC generator
4. Parity checker
5. Backoff and intergap timers

For more information on the transmit block see Chapter 6, “MAC Operation.”

MAC Receive Block

The receive block is responsible for receiving data. It complies with the IEEE 802.3 Standard for Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. It also supports a full-duplex mode of operation, which allows simultaneous transmission while receiving.

The receive block consists of the following sections:

1. Receive FIFO, FIFO controller, and counters
2. CAM block for address recognition
3. CRC generator and checker
4. Parity generator

For more information on the receive block see Chapter 6, “MAC Operation.”

Flow Control Block

The flow control block provides for the following functions:

- Recognition of MAC Control frames received by the receive block.
- Transmission of MAC Control frames, even if transmitter is paused.
- Timers and counters for PAUSE operation.
- Command and Status Register interface.
- Options for passing MAC Control frames through to S/W drivers.

The receive logic in the flow control block recognizes a MAC Control frame and performs the PAUSE Operation as follows: First, the length/type field must have the special value specified for MAC Control frames. Second, the destination address must be recognized by the CAM. Third, the frame length must be 64 bytes, including CRC. Fourth, the CRC must be good. And fifth, the frame must contain a valid PAUSE operation code and operand value.

If the length/type field does not have the special value specified for MAC Control frames, then the MAC takes no action, and the packet is treated as a normal packet. If the CAM does not recognize the destination address, the MAC rejects the packet. If the packet length is not 64 bytes, including the CRC, the MAC will not perform the operation. The packet will be marked as a MAC Control packet, and passed forward to the S/W drivers, if pass through is enabled.

A control bit in the Transmit Status register can be set to generate a Full Duplex PAUSE Operation or other MAC Control function, even if the transmitter itself is paused.

There are two timers and corresponding CSR registers which are used during PAUSE operation. One timer and register are used when a received packet causes the transmitter to be paused. The other pair is used to approximate the paused status of the other end of the link, after the transmitter sends a PAUSE command.

The Command and Status Register (CSR) interface provides control and status bits within the transmit and receive control registers and status registers. These allow the initiation of sending a MAC Control frame, enabling and disabling MAC Control functions, and reading of the Flow Control counters.

Control bits are provided for either processing MAC Control frames entirely within the controller, or for passing MAC Control frames on the S/W drivers. This allows Flow Control to be enabled by default even if S/W drivers which are not otherwise "Flow Control aware".

MAC Control and Status Registers

The MAC has a block of registers which provides control and status information. These registers control the transmit and receive blocks, report MAC status, and provide a communication interface to the CAM, the MII station management interface, and the optional external EEPROM/ROM. These registers are available via PCI memory-mapped or I/O mapped access. For more information on the MAC control and status registers see the section “MAC Layer Registers” in Chapter 4.

MII Station Manager

The Toshiba 100/10Mbps Ethernet MAC processes the station Management Data (MDIO and MDC) signals of the MII controller, but does not interpret the values. The MII station manager provides logic for reading and writing control and status registers in a configured PHY device through the MII defined serial interface.

If a specialized application, such as a bridge, router, or switched hub, requires access to these values to negotiate configurations, they are available via MAC control and status registers which trigger reads and writes across the station Management Data interface. For more information see the section “Station Management Data Access Registers” in chapter4.

PROM Controller

The PROM controller provides logic for reading and writing an optional external EEPROM or ROM device. The external devices supported are the MicroChip 93LC46B and the National NM93C46. Timing compatible devices, smaller devices, and Read-only equivalent devices are also supported. For more information see the section “PROM Control Registers” in Chapter 4.

Overview

This chapter describes user accessible registers for the TC35815CF Flow Control 10/100Mbps Ethernet Controller. Registers are grouped by function:

- PCI configuration
- DMA control and status registers
- Flow Control registers
- MAC control and status

In normal operation, most registers do not need to be accessed directly. Transmit and receive operations are done using continuous cyclic queues, or rings. Control and status information is communicated through data structures described in Chapter 5. DMA control registers need to be accessed at initialization time, in order to initiate operation. MAC registers may need to be accessed for special configuration needs, such as setting the CAM to do address filtering. For interrupt based drivers, some of the DMA and MAC registers are accessed in the interrupt handler, to enable and disable interrupts, to determine the cause of an interrupt, and to clear interrupt condition bits. Flow control registers may be accessed by drivers to monitor the progress of local and remote PAUSE commands.

Table 4-1 shows the register name, symbol, address, size, and access type for each PCI configuration register.

Table 4-1 PCI Configuration Registers

Name	Symbol	Addr	Bytes	Access
Vendor ID	Vend_ID	00h	2	RO
Device ID	Dev_ID	02h	2	RO
Command	PCI_Cmd	04h	2	R/W
Status	PCI_Stat	06h	2	R/W
Class Code	PCI_Clas	08h	4	RO
PCI Control	PCI_Ctl	0Ch	4	R/W*
I/O Base Address	IO_BaseA	10h	4	R/W
Memory Base Address	MLo_BaseA	14h	4	R/W
Subsystem Vendor ID	Sub_Vend_ID	2Ch	2	R/W
Subsystem ID	Sub_ID	2Eh	2	R/W
PCI Interrupt	PCI_Int	3Ch	4	R/W*

The PCI configuration registers implement a standard master/slave PCI device. They provide device identification, control and status information, and various setup registers, which are addressed at initialization time.

PCI configuration registers can be loaded from an EEPROM/ROM or initialized by system driver code which asserts the IDSel signal.

Table 4-2 shows the register name, symbol, address, size, and access type for each DMA control register. These registers provide support for a transmit queue, a receive queue, free buffer list, and a free descriptor area. Some registers control fragmentation sizes or polling rates.

Table 4-2 DMA Control Registers

Name	Symbol	Addr	Bytes	Access
DMA Control	DMA_Ctl	00h	3	R/W
Transmit Frame Pointer	TxFrmPtr	04h	4	R/W
Transmit Threshold	TxThrsh	08h	2	R/W
Transmit Polling Counter	TxPollCtr	0Ch	2	R/W
Buffer List Frame Pointer	BLFrmPtr	10h	4	R/W
Receive Fragment Size	RxFragSize	14h	2	R/W
Interrupt Enable	Int_En	18h	2	R/W
Free Descriptor Area Base	FDA_Bas	1Ch	4	R/W
Free Descriptor Area Limit	FDA_Lim	20h	2	R/W
Interrupt Source	Int_Src	24h	2	R/W*

Table 4-3 shows the register name, symbol, address, size, and access type for each Flow Control register. These registers provide support for receiving, processing, and generating Full Duplex PAUSE Operations and other forms of MAC Control packets.

Table 4-3 Flow Control Registers

Name	Symbol	Addr	Bytes	Access
Pause Count	PauseCnt	30h	2	RO
Remote Pause Count	RemPauCnt	34h	4	RO
Transmit Control Frame Status	TxConFrmStat	38h	2	R/W

Table 4-4 shows the functional group, register name, symbol, address, size, and access type for each MAC control and status register. These registers are generally loaded from an EEPROM or ROM on chip power up or chip reset. Some of these registers, such as the CAM control and error count registers, are accessed by system software drivers while the MAC is active. The MAC transmit and receive control and status registers are generally controlled by the DMA engine after they are setup.

Table 4-4 MAC Control and Status Registers

Group	Name	Symbol	Addr	Bytes	Access
MAC	MAC Control	MAC_Ctl	40h	2	R/W*
	CAM Control	CAM_Ctl	44h	1	R/W
	Transmit Control	Tx_Ctl	48h	2	R/W*
	Transmit Status	Tx_Stat	4Ch	2	RO
	Receive Control	Rx_Ctl	50h	2	R/W
	Receive Status	Rx_Stat	54h	2	RO
Station Management Data	Station Management Data	MD_Data	58h	2	R/W
	Station Management Control and Address	MD_CA	5Ch	2	R/W*
CAM	CAM Address	CAM_Adr	60h	2	R/W
	CAM Data	CAM_Data	64h	4	R/W
	CAM Enable	CAM_Ena	68h	3	R/W
EEPROM	PROM Control	PROM_Ctl	6Ch	2	R/W*
	PROM Data	PROM_Data	70h	2	R/W
System Management	Missed Error Count	Miss_Cnt	7Ch	2	RClr/W

The MAC layer control registers include a master MAC control register, control registers for transmit and receive, control registers for the CAM, and some error counters for network management.

Register Access

There are three basic types of register access:

1. RO - Read Only. Writing to a RO register has no effect.

2. R/W - Read and Write. The user can read and write the register. The controller may also write the register, and some parts may be reserved or read only, so the user may not always be able to read what was written.

3. RClr/W - Read with Clear and Write. Reading the register clears the value. Used for counters. May also be written to change the count period.

The registers with access marked with “*” have special semantics, such as bits which are “write 1 to clear”, etc. as explained in the detailed descriptions.

Reserved bits are initialized to 0. Software should leave them unchanged when writing to registers, to aid in compatibility with future uses. Software should not depend on the value of reserved fields being 0.

Register Address Summary

Figures 4-1 through 4-4 summarize the memory addresses of the PCI configuration registers, the DMA control and status registers, Flow Control registers, and MAC control and status registers. These figures assume a Little Endian layout of system memory.

	Byte 3	Byte 2	Byte 1	Byte 0	
	Dev_ID		Vend_ID		00
	PCI_Stat		PCI_Cmd		04
PCI_Clas	Base_Cl	Sub_Cl	Prog_IF	Rev_ID	08
PCI_Ctl	BIST	Hdr_Typ	Lat_Timr	Cache_Sz	0C
	IO_BaseA				10
	MLo_BaseA				14
	Reserved (MHi_BaseA)				18
	Reserved				1C
	Reserved				...
	Sub_ID		Sub_Vend_ID		2C
	Reserved				...
PCI_Int	Max_Lat	Min_Gnt	Int_Pin	Int_Line	3C

Figure 4-1 Address Map of PCI Configuration Registers

Byte 3	Byte 2	Byte 1	Byte 0	
Reserved		DMA_Ctl		00
TxFrmPtr				04
Reserved		TxThrsh		08
Reserved		TxPollCtr		0C
BLFrmPtr				10
Reserved		RxFragSize		14
Reserved		Int_En		18
FDA_Bas				1C
FDA_Lim				20
Reserved		Int_Src		24

Figure 4-2 Address Map of DMA Control Registers

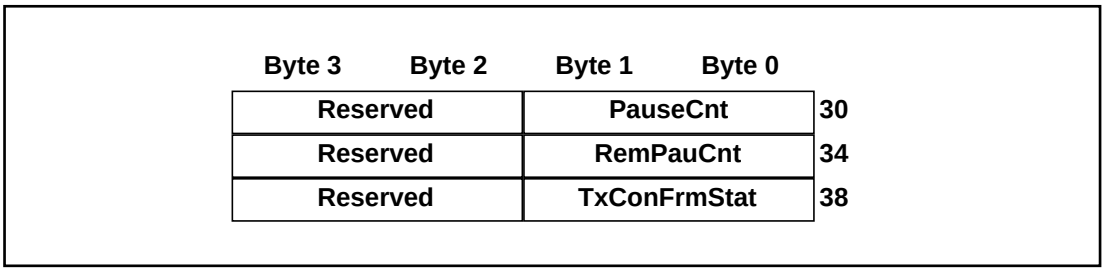


Figure 4-3 Address Map of Flow Control Registers

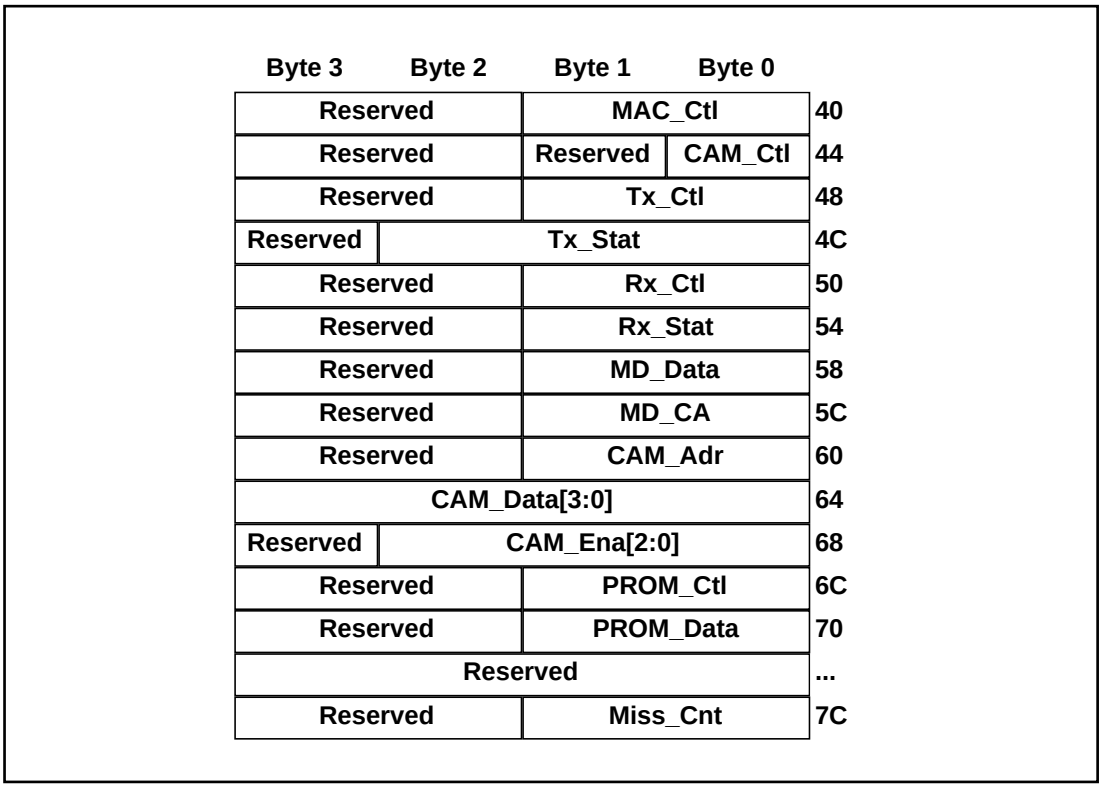


Figure 4-4 Address Map of MAC Control and Status Registers

The PCI Configuration Register addresses are effective when the IDSel signal is asserted. The DMA and MAC Control and Status Register addresses are effective when IDSel is not asserted and when the upper bits of the address match either the I/O base register (IO_BaseA) or the memory base register (MLo_BaseA).

PCI Configuration Registers

There are a total of 256 configuration address spaces reserved for a PCI device. The first 64 address spaces are predefined by the PCI specification. The TC35815CF Flow Control 10/100Mbps Ethernet Controller uses only the first 64 address spaces.

Vendor ID Register

Vendor ID

(Vend_ID) **00h**

15	0
Vendor ID number	

The vendor ID register identifies TOSHIBA as the component manufacturer. It contains a unique identification number assigned by PCI SIG. The Toshiba PCI SIG number is 102Fh.

Device ID Register

Device ID

(Dev_ID) **02h**

15	0
Device ID number	

The PCI Device ID register identifies the specific device. The 16-bit number is assigned by Toshiba is 0030h. Revisions are part of the Class Code register.

PCI Command Register

PCI Command

(PCI_Cmd) 04h

15	10	9	8	7	6	5	4	3	2	1	0
Reserved		FastEn	SErrEn	WaitCC	ParER	VGA_PS	MWIEn	SpecC	BusMas	MemS	IOS

FastEn	Fast Back-to-Back Enable	=0. Controller cannot do fast back-to-back transactions to different devices.
SErrEn	System Error Enable	Enable the system error (SERR#) driver.
WaitCC	Wait Cycle Control	=0. Controller does not generate address/data stepping.
ParER	Parity Error Response	Device is to respond to parity errors.
VGA_PS	VGA Palette Snoop	=0. No special VGA palette snooping.
MWIEn	Memory Write & Invalidate Enable	=0. Controller does not generate Memory Write & Invalidate commands.
SpecC	Special Cycles	=0. Controller ignores special cycle operations.
BusMas	Bus Master	Device can behave as a bus master.
MemS	Memory Space	Device can respond to memory accesses.
IOS	I/O Space	Device can respond to I/O space accesses.

This register describes how the controller can generate or respond to PCI cycles. For a more complete description, see the PCI Specification, Section 6.2.2, Device Control.

The following are required for the Ethernet Controller to operate correctly: BusMas, and either MemS or IOS.

The following are optional for the Ethernet Controller: SErrEn, and ParER.

The following are always 0 and ignored by the Ethernet Controller: FastEn, WaitCC, VGA_PS, MWIEn, and SpecC.

The controller only contains one address comparator, which is shared between memory and I/O accesses as follows: when MemS=0 and IOS=1, the controller will respond to any accesses; when MemS=1 the controller will respond to memory space accesses only, regardless of the value of IOS. In addition to MemS and IOS, the corresponding Base Address register must be set.

Hardware reset to value 000h. Software reset has no effect.

Status Register

Status

(PCI_Stat) **06h**

15	14	13	12	11	10 9	8	7	6 0
DParErr	SSysErr	RMasAbt	RTarAbt	STarAbt	DEVSEL	DParD	FastCap	Reserved

DParErr	Detected Parity Error	Parity error was detected. Set even if Parity Error Response bit in PCI Command register is not set. Parity errors during address cycles are included.
SSysErr	Signaled System Error	Set when device asserts the SERR# line.
RMasAbt	Received Master Abort	This device was master and a master transaction was terminated with master-abort, except for a special cycle.
RTarAbt	Received Target Abort	This device was master and a master transaction was terminated with target-abort.
STarAbt	Signaled Target Abort	This device was target and a master transaction was terminated by asserting target-abort.
DEVSEL	Device selection timing	=01. The controller's slowest DEVSEL# timing as a target device is "medium". The bits are encoded: 00 - Fast 01 - Medium 10 - Slow
DParD	Data Parity Detected	Set when (1) PERR# asserted or observed by the controller, (2) agent setting PERR# acted as master, and (3) Parity Error Response bit in PCI Control register is set.
FastCap	Fast Back-to-Back Capable	=0. The controller is not capable of accepting fast back-to-back transactions from different agents.

This register records status information for PCI bus related events. This register has some special operating characteristics. For example, individual bits are cleared by writing a 1; writing a 0 has no effect. This is necessary to support simultaneous updating of status information by both the system and the PCI device. For a more complete description, see the PCI Specification, Section 6.2.3, Device Status.

Software reset to value 0200h.

Class Code Register

Class Code

(PCI_Clas) **08h**

31 — 24	23 — 16	15 — 8	7 — 0
Base_Cl	Sub_Cl	Prog_IF	Rev_ID

Base_Cl	Base Class ID	Set to 02h for Network controller.
Sub_Cl	Sub Class ID	Set to 00h for Ethernet controller.
Prog_IF	Programming Interface	Set to 00h. No register level programming interfaces defined.
Rev_ID	Revision ID	Set by manufacturer to a device specific revision number, an extension to the Device ID register.

PCI Control Register

PCI Control

(PCI_Ctl) **0Ch**

31 — 24	23 — 16	15 — 8	7	0
BIST	Hdr_Typ	Lat_Timr	0	Cache_Sz

BIST	Built-in Self Test	Controls invocation of self test code on start up.
Hdr_Typ	Header Type	Set to 00h for single function, standard layout of bytes 10h through 3Fh in configuration space.
Lat_Timr	Latency Timer	Number of PCI bus clocks for controller as bus master.
Cache_Sz	Cache Line Size	System cache line size.

Hardware reset to value 8000_0000h. Software reset has no effect.

The Built-in Self Test (BIST) is used to test the on-chip buffers. The high order of BIST (bit 31 of PCI_CTL) is a R/O 1, indicating the controller supports BIST. The next bit (bit 30 of PCI_CTL) is used to start the test. Writing a 1 to bit 30 invokes the test; bit 30 is cleared when the test is complete. If the test fails, the low order bits of the BIST status (bits 25, 24 in PCI_Ctl) are set to indicate the nature of the error; bit 25 indicates a RAM parity error; bit 24 indicates a RAM test failure: read data did not match expected data. Bits 29 to 26 are reserved.

BIST should be invoked by software drivers during initialization. Note: invoking BIST will overwrite the RAM based registers.

With a 33MHz clock, the BIST test takes approximately 123µs, assuming a 1K

double-word memory.

The default value of the Latency Timer is 0, to indicate it is programmable.

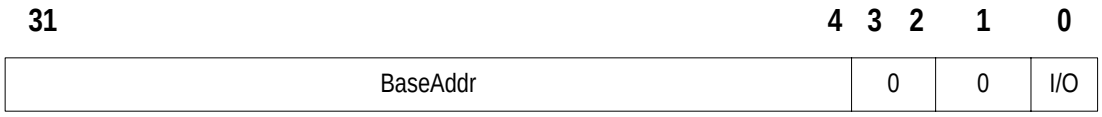
On hardware reset, the Cache Line Size is set to zero. It is the responsibility of the S/W drivers to set an appropriate default cache line size. The recommended size for most systems is eight doublewords (32 bytes). The controller can support cache line sizes up to 127 doublewords. The cache line size is used to select Memory Read Multiple commands instead of Memory Read Line commands for burst reads.

I/O and Memory Base Address Registers

These registers are used to map the DMA and MAC control and status registers into either I/O address space or system memory space. I/O address space and memory address space are both limited to 32 bits. In addition to setting a base address register, the corresponding control bit in the PCI Command register must be set.

I/O Base Address

(IO_BaseA) 10h

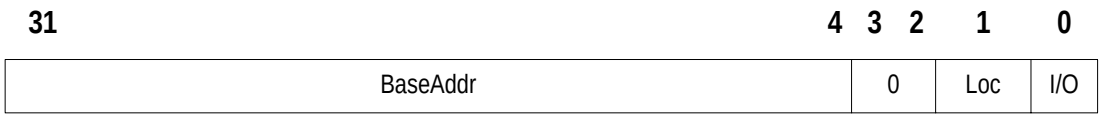


BaseAddr	Base Address	Upper 28-bits of base address.
I/O	I/O Flag	1= I/O space (Read-Only).

Hardware reset to value 0000_0001h. Software reset has no effect.

Memory Base Address

(MLo_BaseA) 14h



BaseAddr	Base Address	Upper 28-bits of base address.
Loc	Location Bits	00= locate anywhere in 32-bit address space (Read-Only).
I/O	I/O Flag	0= Memory space (Read-Only).

Hardware reset to value 0000_0000h. Software reset has no effect.

Subsystem Vendor ID Register

After hardware reset, the Subsystem Vendor ID and Subsystem ID registers are initialized with values read from the EEPROM/ROM, if present. An attempt to read these registers while they are being loaded will be terminated with a target retry.

Subsystem Vendor ID Register

(Sub_Vend_ID) **2Ch**

15	0
Subsystem Vendor ID number	

The subsystem vendor ID register identifies the adapter card manufacturer. It contains a unique identification number assigned by PCI SIG. The default value is 0. If an EEPROM is present, it can be used to set the value.

Subsystem ID Register

Subsystem ID

(Sub_ID) **2Eh**

15	0
Subsystem ID number	

The subsystem ID register identifies the specific device. The 16-bit number is assigned by the adapter card manufacturer. Vendors that produce both devices and subsystems must ensure that Subsystem ID values are different from Device ID values. The default value is 0. If an EEPROM is present, it is used to set the value after hardware reset.

PCI Interrupt Register

PCI Interrupt

(PCI_Int) **3Ch**

31 24	23 16	15 8	7 0
Max_Lat	Min_Gnt	Int_Pin	Int_Line

Max_Lat	Maximum Latency	RO	=0 (unit = 1/4 microsecond) Desired setting for latency timer vales.
Min_Gnt	Minimum Grant	RO	= 0 (unit = 1/4 microsecond) Minimum burst period, assuming 33MHz clock.
Int_Pin	Interrupt Pin	RO	Set to 01h for INTA#.
Int_Line	Interrupt Line	R/W	Set by system. Interrupt line routing information.

The value of 0 for Max_Lat and Min_Gnt indicate that these registers are not used for determining these values.

To determine if an interrupt is being generated by this PCI device, the S/W should examine the Interrupt Source register. For more information see Chapter 4, page 49, in the section “Interrupt Source Register.”

DMA Control Registers

There are three queues which are jointly managed by the DMA engine and the system software:

1. Transmit Queue - TxQ
2. Receive Queue - RxQ
3. Buffer List -BL_Q

The transmit queue is a list of frame descriptors which are ready for transmission. The receive queue is a list of frame descriptors which have been received and are ready for processing by the system software. The buffer list is a list of buffer descriptors which describes areas of system memory that can be used to store received data. The free descriptor area (FDA) is the memory area where the controller writes the frame descriptors and buffer descriptors for the Receive Queue.

DMA Control

DMA Control

(DMA_Ctl) 00h

31 19	18	17	16	15	14	13	12	11 9	8 2	1	0
Reserved	IntMask	SWIntReq	TxWakeUp	RxBigE	TxBigE	TestMode	PowrMgmnt	Reserved	DmBurst	0	0

DmBurst	DMA Burst Size	Size of data bursts requested in master mode.
PowrMgmnt	Power Management	Enable dynamic Power Management.
TestMode	Test Mode	Enable Test Mode functions.
TxBigE	Transmit Big Endian	If set, treat all transmit data as big endian.
RxBigE	Receive Big Endian	If set, treat receive data as big endian.
TxWakeUp	Transmit Wake Up	When set, abort the current polling cycle and begin transmission.
SWIntReq	Software Interrupt Request	When set, cause an interrupt to be signalled.
IntMask	Interrupt Mask	When set, causes interrupt signals to be disabled.

Hardware reset to value 0000_1020h. Software reset has no effect.

The DMA control register controls the transfer of data in master mode: burst size, big endian byte ordering, and test mode functionality. It also controls a number of other DMA functions, such as power management, wake-up on transmit, and software interrupt.

The DmBurst field controls the size of data transfers requested across the PCI Bus when in master mode. It is a nine-bit register, with the two low-order bits forced to zero, i.e. values must be a multiple of four. The default value after hardware reset is 32 bytes or 8 doublewords. This can be modified by a value in the EEPROM/ROM. DmBurst cannot be set to zero; an attempt to write a zero is ignored. Generally, the DmBurst register should indicate a multiple of the PCI cache line size register. Care must be used with burst sizes of 4, 8, and 12 in 100Mbps full duplex mode.

The Power Management bit controls dynamic power management. The default value is 1, or enabled.

TestMode enables certain test features, such as the ability to read and write all of internal DMA RAM.

The Transmit and Receive Big Endian bits support the transmission and reception of data which has been ordered for a Big Endian machine. Note that only data (bytes in the areas pointed to by the buffer descriptors) are affected. Control information, which includes registers, frame descriptors, and buffer descriptors, are always in native PCI bus, or Little Endian format.

The Transmit Wake Up bit supports immediate transmission of data, rather than waiting for the current polling cycle to end. If set, and if the transmitter is polling, the current polling cycle will be terminated. This bit is cleared at the end of the current polling cycle.

Software interrupt is provided by the controller as a service for software drivers.

Transmit Frame Pointer

Transmit Frame Pointer

(TxFrmPtr) 04h

EOL	End of List	If set, Address is invalid for use by the controller. Must wait for system to clear.

Software reset to value 0000_0001h.

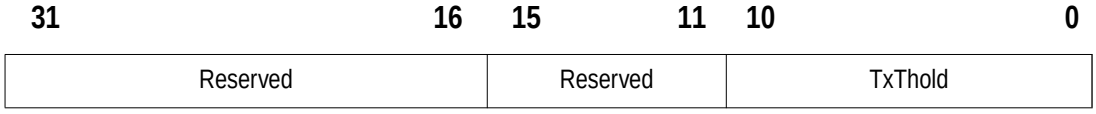
The transmit frame pointer register contains the address of the first frame descriptor to transmit. After software reset, the EOL bit is set. The system must set this register to a properly initialized frame descriptor to enable transmission, or polling for packets to transmit. A valid address must be aligned to a 16 byte boundary, i.e. bits 0 to 3 must be zero.

For a description of how polling is controlled, see section "Transmit Polling Control Register".

Transmit Threshold Control

Transmit Threshold

(TxThrsh) 08h



Held in internal RAM, so hardware and software reset have no effect.

The transmit threshold register controls the buffer latency for transmitting packets. If the threshold value is non-zero, then data transfer to the MAC will begin as soon as the DMA transmit FIFO contains this number of bytes, or as soon as a complete packet is in the FIFO. If the threshold value is zero, data transfer to the MAC starts immediately. It is the responsibility of the S/W drivers to initialize this register. The value can be specified in the EEPROM/ROM.

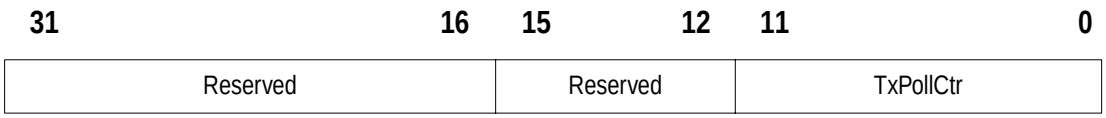
If the threshold value is set too low, the DMA transmit FIFO may run dry due to PCI bus latency. If this occurs, as indicated by the MAC transmit status, system software should increase the TxThold value.

Care should be taken not to set the threshold value greater than 1700. In long packet mode, this may cause buffer memory to fill without enabling transmission, causing the transmitter to hang.

Transmit Polling Control Register

Transmit Polling Counter

(TxPollCtr) 0Ch



Held in internal RAM, so hardware and software reset have no effect.

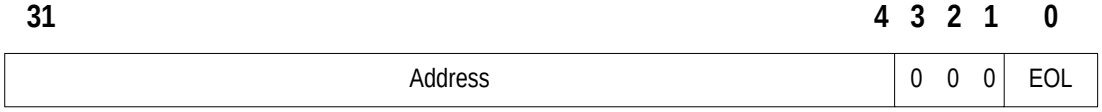
The transmit polling counter controls the frequency with which the controller polls for packets to transmit. An internal counter will be set to this value, and decrements to zero. When the register reaches zero, a read will be done to see if a new transmit packet has arrived. With a 33MHz clock, each unit in the polling counter is equivalent to 61.44 micro-seconds. The transmit polling counter is held in internal RAM, so does not have a default value. It is the responsibility of the S/W drivers to initialize this register. The value can be specified in the EEPROM/ROM, with software driver support.

This register must be set to 1 or higher, if "hot chaining" of Transmit buffers is used.

Buffer List Frame Pointer

Buffer List Frame Pointer

(BLFrmPtr) 10h



EOL	End of List	If set, Address is invalid for use by the controller. Must wait for system to clear.

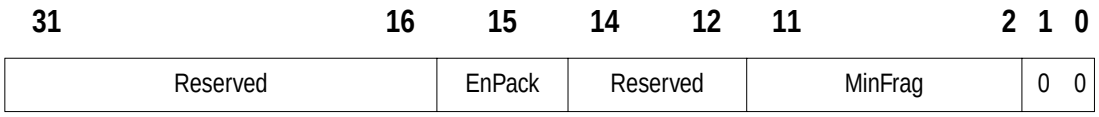
Software reset to value 0000_001h.

The buffer list frame pointer contains the address of the first frame descriptor to read for acquiring free buffer descriptors. The system must set this register to a properly initialized frame descriptor to enable reception. A valid address must be aligned to a 16 byte boundary, i.e. bits 0 to 3 must be zero.

Receive Fragment Size Register

Receive Fragment Size

(RxFragSize) 14h



MinFrag	Minimum Fragment	Minimum number of bytes to write into a partially filled buffer.
EnPack	Enable Packing	1= Use MinFrag value to pack buffers. 0= Use FDCtl field to control packing (Default).

Hardware reset to value 0000_0000h. Software reset has no effect.

The receive fragment size register specifies the smallest data fragment the controller will generate. The size must be a multiple of four, i.e. the two low order bits are always zero. Packing can be enabled globally, using the Enable Packing bit, or on a per buffer area basis, as explained in Chapter 5, page 65 in the section “Frame Descriptor Control Field (FDCtl).”

The controller always begins storing received data on a four-byte-aligned address. There may be one to three unused bytes at the end of a frame, due to alignment.

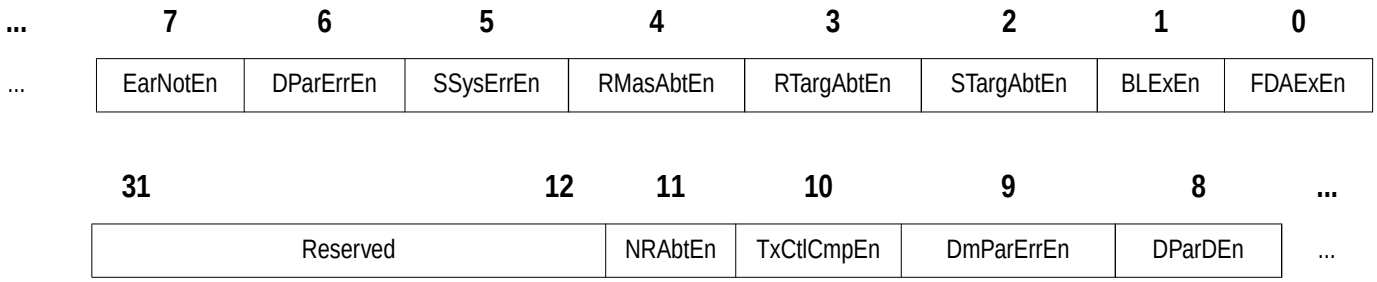
When packing is enabled, the MinFrag value must be greater than zero for the controller to work. It is the responsibility of S/W drivers to set the Minimum Fragment field and the EnPack bit if packing is desired.

When packing is not enabled, the MinFrag value must be left at zero.

Interrupt Enable Register

Interrupt Enable

(Int_En) **18h**



FDAExEn	Free Descriptor Area Exhausted Enable	Enable an interrupt if the Free Descriptor Area becomes exhausted, i.e. if the controller encounters a block in the FDA which is still owned by the system.
BLExEn	Buffer List Exhausted Enable	Enable an interrupt if the Buffer List becomes exhausted, i.e. if the controller encounters a descriptor in the BL which is still owned by the system.
STargAbtEn	Signalled Target Abort Enable	Enable an interrupt if the controller signals a target abort while acting as a target.
RTargAbtEn	Received Target Abort Enable	Enable an interrupt if the controller receives a target abort while acting as bus master.
RMasAbtEn	Received Master Abort Enable	Enable an interrupt if the controller receives a master abort while acting as a target.
SSysErrEn	Signalled System Error Enable	Enable an interrupt if the controller signals system error.
DParErrEn	Detected Parity Error Enable	Enable an interrupt if the controller detects a parity error on a PCI bus data transfer during a master access.
EarNotEn	Early Notify Enable	Enable an interrupt after writing the first buffer and buffer descriptor of a packet.
DParDEn	Data Parity Detected Enable	Enable an interrupt if bit 8 of PCI Stat register is set.
DmParErrEn	DMA Parity Error Enable	Enable an interrupt if a parity error is detected in reading or writing the DMA internal RAM.
TxCtlCmpEn	Transmit Control Complete Enable	Enable an interrupt when transmission of a MAC Control packet is complete
NRAbtEn	Non-recoverable Abort Enable	Enable an interrupt when there is an internal non-recoverable abort condition.

Hardware reset to value 0000_0000h. Software reset has no effect.

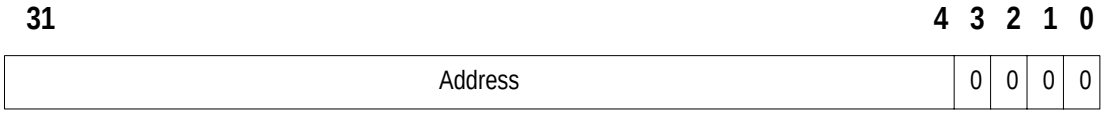
The interrupt enable register controls the generation of interrupts in response to errors and other conditions detected by the DMA engine.

The Early Notify enable bit supports applications which want to minimize latency. Note that the Frame Descriptor will not be valid when the Early Notify is processed. Only the first buffer descriptor is valid when this interrupt is signalled.

Descriptor Area Registers

Free Descriptor Area Base Register

(FDA_Bas) 1Ch

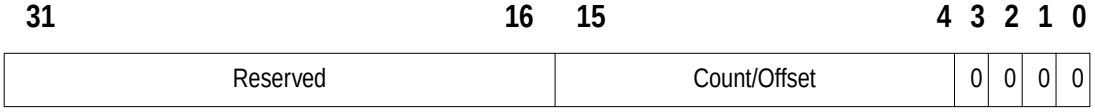


Hardware reset to value 0000_0000h. Software reset has no effect.

The free descriptor area base register contains the starting address of the area reserved for the controller to write frame and buffer descriptors for received packets. The address must be a multiple of 16 bytes, i.e. bits 0, 1, 2, and 3 are zero.

Free Descriptor Area Limit Register

(FDA_Lim) 20h



Hardware reset to value 0000_0000h. Software reset has no effect.

The free descriptor area limit register contains the count of the number of 16-byte blocks in the receive descriptor area, in bits 15:4. Alternatively, the low 16 bits can be viewed as a byte offset from the base.

Each 16-byte block holds one frame descriptor, or two eight-byte buffer descriptors. So the maximum size of a single descriptor area is 4095*16 or 64K - 16 bytes.

Note: The FDA_Lim register must point to the lowest offset in the FDA where a new frame descriptor can be safely begun. Enough space must be allowed for a maximum size packet to have enough space for a maximum number of buffer descriptors.

Interrupt Source Register

Interrupt Source

(Int_Src) **24h**

31 15	14	13	12	11	10	9	8	7
Reserved	NRAbt	DmParErrStat	BLEx	FDAEx	IntNRAbt	IntTxCtlCmp	IntExBD	DmParErr

6	5	4	3	2	1	0
IntEarNot	SWInt	IntBLEx	IntFDAEx	IntPCI	IntMacRx	IntMacTx

IntMacTx	Interrupt reported in MAC Transmit Status, Tx_Stat.	W1/Clr
IntMacRx	Interrupt reported in MAC Receive Status, Rx_Stat.	W1/Clr
IntPCI	Interrupt reported in PCI Controller, PCI_Stat	R/O
IntFDAEx	Interrupt caused by Free Descriptor Area Exhausted	R/O
IntBLEx	Interrupt caused by Buffer List Exhausted	R/O
SWInt	Interrupt caused by Software Interrupt Request	R/O
IntEarNot	Interrupt caused by Early Notify	W1/Clr
DmParErr	Interrupt caused by DMA Parity Error.	R/O
IntExBD	Interrupt caused by Excessive (more than 28) Buffer Descriptors.	W1/Clr
IntTx/CtlCmp	Interrupt caused by MAC Control packet completed.	W1/Clr
IntNRAbt	Interrupt caused by Non-recoverable Abort state.	R/O
FDAEx	Set if FDA is exhausted. Cleared by writing 1.	W1/Clr
BLEx	Set if BL is exhausted. Cleared by writing 1.	W1/Clr
DmParErrStat	Set if DMA Parity Error occurs. Cleared by writing 1.	W1/Clr
NRAbt	Set if Non-recoverable abort occurs.	W1/Clr

W1/Clr- indicates a bit that is cleared by writing 1 to the bit. Writing 0 has no effect.
RO- indicates a read only bit, which is cleared by clearing the condition which sets the bit, or by resetting the controller.

Software reset to value 0000_0000h.

The Interrupt Source Register is read by system software, to see if there is an interrupt associated with the Ethernet Controller. In addition, the register provides status bits for some conditions which are not reported elsewhere.

If an interrupt is associated with the Ethernet Controller, all further interrupts from the Ethernet Controller can be masked with the IntMask bit of the DMA Control register.

If bits 10 through 0 are all zeroes, the controller did not generate the interrupt. Low order bits are set if interrupts are generated and reported in other status registers.

Bit 8 is set if a single Frame Descriptor requires more than 28 Buffer Descriptors. Bit 13 is set whenever a DMA RAM parity error is detected. Bit 7 is set an interrupt is generated, only if the DParErrEn bit of the Interrupt Enable Register is set.

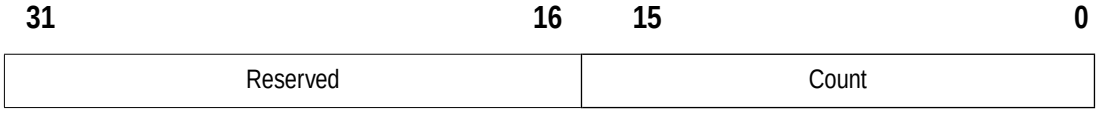
Flow Control Registers

Pause Control

(PauseCnt) 30h

Remote Pause Control

(RemPauCnt) 34h



PauseCnt	Received Pause Count	Count of slot times that Transmitter is being paused, as the result of receiving a MAC Control PAUSE operation packet.
RemPauCnt	Remote Pause Count	Count of slot times that remote MAC is being paused, as the result of sending a PAUSE operation packet.

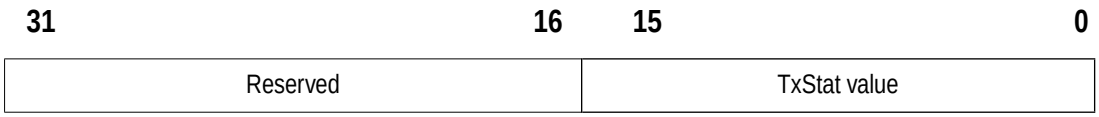
Software reset to value 0000_0000h.

The pause count register provides the current value of a received pause counter. A value of 0 indicates the MAC is not paused. The remote pause count register provides an approximate current value of the remote pause counter, based on when a PAUSE command was sent.

For both counters each unit is one slot time, or 512 bit times.

Transmit Control Frame Status

(TxCtlFrmStat) 38h



Held in internal RAM, so hardware and software reset have no effect.

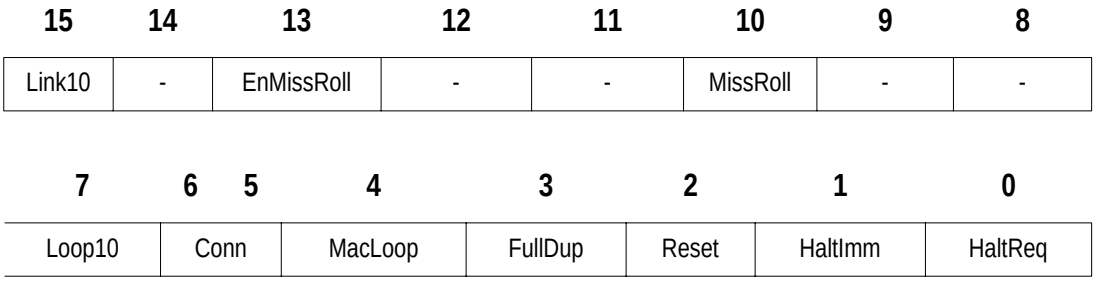
The Transmit Control Frame Status provides the status of sending a MAC Control packet to a remote station via the SdPause bit of the Transmit Control register. When the transmission is complete, the software drivers may pick up the status from this register. Software can reset this register by clearing it before initiating the transfer of a MAC Control frame. The TxCtlCmpEn bit of the Interrupt Enable register provides for an option to generate an interrupt on completion of MAC Control packet transmission.

MAC Layer Registers

MAC Control Register

MAC Control

(MAC_Ctl) 40h



HaltReq	Halt Request	Stop transmission and reception after completion of any current packets.
HaltImm	Halt Immediate	Stop transmission and reception immediately.
Reset	Software Reset	Reset all Ethernet Controller state machines and FIFOs.
FullDup	Full Duplex	Allow transmission to begin while reception is occurring.
MacLoop	MAC Loopback	Cause transmission signals to be presented as input to the receive circuit without leaving the controller.
Conn	Connection Mode	Select the connection mode. 00 = Automatic, (default) 01 = Force 10Mbps endec 10 = Force MII (rate determined by MII clock)
Loop10	Loop 10 Mbps	If set, assert the Loop_10 external signal to the 10Mbps endec.
MissRoll	Missed Roll	Missed error counter rolled over.
EnMissRoll	Enable Missed Roll	Interrupt when missed error counter rolls over.
Link10	Link Status 10Mbits	Buffered signal on the Link 10 pin.

Hardware reset to value 0000h. Software reset is invoked by setting bit 2, Reset. Bit 2 is cleared after software reset is complete. Other bits are not affected by software reset.

Software reset is delayed for three clock cycles, to allow normal completion of the operation which writes the Reset bit. Software can use the Tx_ctl and Rx_ctl registers to request halt after current network transactions are complete before using reset.

The Missed Roll and Link Status 10 Mb/s are Read-only status bits. All others are control bits. The Missed Roll bit is set when the counter rolls over and cleared when S/W reads the Missed Count register, as described in section, "System Error Count Register."

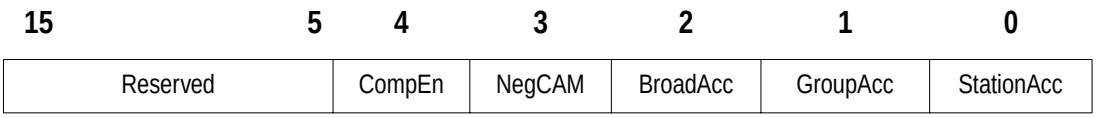
Some PHYs may not support full duplex. MAC Loopback overrides the full duplex bit. Some 10 Mb/s PHYS may interpret Loop10 to control different functions, and signal Link10 to indicate a different status condition.

In automatic connect mode, activity in the form of receive clock and carrier sense on the 10 Mb/s interface will select the 10 Mb/s endec, otherwise the MII is selected. For some full duplex operating environments, the 10 Mb/s 7-wire interface may require software configuration via the Connection Mode bits.

CAM Control Register

CAM Control

(CAM_Ctl) 44h



StationAcc	Station Accept	Accept any packet with a “unicast” station address.
GroupAcc	Group Accept	Accept any packet with a multicast-group address.
BroadAcc	Broadcast Accept	Accept any packet with a broadcast address.
NegCAM	Negative CAM	0 = Accept packets CAM recognizes, reject others. 1 = Reject packets CAM recognizes, accept others.
CompEn	Compare Enable	Enable compare mode.

Hardware reset to value 0000h. Software reset has no effect.

The three accept bits override CAM rejections.

To place the MAC in promiscuous mode, accept packets with all three types of destination address:

1. Station, which has an even first byte, for example, 00-00-00-00-00-00,

2. Multicast-group, which has an odd first byte, but which is not ff-ff-ff-ff-ff-ff, for example, 01-00-00-00-00-00,

3. Broadcast, defined to be ff-ff-ff-ff-ff-ff.

When the CAM compare mode is enabled, the CAM memory is read for addresses to filter incoming messages. The CAM memory is organized as entries of six bytes each, which can be individually enabled and disabled as described in section “CAM Access Registers”, on page 59.

An alternative way to place the MAC in promiscuous mode is to set the Negative CAM bit, but clear the Compare Enable bit. This way, the CAM will fail to recognize all packets, and in turn the MAC will accept them.

To reject all packets, clear all bits in CAM_ctl.

Transmit Control and Status Registers

Transmit Control

(Tx_Ctl) 48h

15	14	13	12	11	10	9
-	EnComp	EnTxPar	EnLateColl	EnExColl	EnLCarr	EnExDefer

8	7	6	5	4	3	2	1	0
EnUnder	MII10	SdPause	NoExDef	FBack	NoCRC	NoPad	TxHalt	TxEn

TxEn	Transmit Enable	If zero, stop transmission immediately.
TxHalt	Transmit Halt Request	Halt transmission after completing any current packet.
NoPad	Suppress Padding	Do not generate pad bytes for packets with less than 64 bytes.
NoCRC	Suppress CRC	Do not add the CRC at the end of a packet.
FBack	Fast Back-off	Use faster back-off timers for testing.
NoExDef	No Excessive Defer	Suppress the checking of Excessive Deferral.
SdPause* MII10	Send Pause MII 10 Mb Mode	Send a PAUSE command, or other MAC Control frame. Set by S/W to enable SQE checking in MII 10 Mb mode.

Interrupt Enable Flags

EnUnder	Enable Underrun	Interrupt if the MAC transmit FIFO becomes empty during transmission.
EnExDefer	Enable Excessive Deferral	Interrupt if the MAC defers for MAX_DEFERRAL time: = 0.24288 ms for 100Mbps. = 2.4288 ms for 10Mbps.
EnLCarr	Enable Lost Carrier	Interrupt if carrier sense is not detected or is dropped during the entire transmission of a packet.
EnExColl	Enable Excessive Collision	Interrupt if 16 collisions occur in the same packet.
EnLateColl	Enable Late Collision	Interrupt if a collision occurs after 512 bit times (64 byte times).
EnTxPar	Enable Transmit Parity	Interrupt if the MAC transmit FIFO has a parity error.
EnComp	Enable Completion	Interrupt when the MAC transmits or discards one packet.

Hardware reset to value 0000h. Software reset clears the TxEn and does not affect others.

*SdPause is automatically cleared upon completing the transmission of the MAC Control packet. Writing 0 to this bit has no effect.

To receive an interrupt after each packet, set the enable completion and all the MAC error enable bits. Interrupts may also be enabled only for specific conditions.

SQE checking is automatic over the 7-wire interface.

Transmit Status

(Tx_Stat) 4Ch

16	15	14	13	12	11	10	9
SQErr	TxHalted	Comp	TxPar	LateColl	Tx10Stat	LostCrs	ExDefer

8	7		5	4	3	2	1	0
Under	IntTx	Paused	TXDefer	ExColl	TxColl			

TxColl	Transmit Collision Count	Count of the collisions in transmitting a single packet. If 16 collisions occur, TxColl will be zero, and ExColl is set.
ExColl	Excessive Collision	Set if 16 collisions occur in the same packet. Transmission skipped.
TxDefer	Transmit Deferred	Set if transmission of packet was deferred and no collisions occurred.
Paused	Transmitter Paused	Set if transmission was paused after completion of the current packet.
IntTx	Interrupt on Transmit	Set if transmission of packet caused an interrupt condition. This includes the EnComp transmission complete condition, if enabled.

Transmission Status Flags

Under	Underrun	MAC transmit FIFO becomes empty during transmission.
ExDefer	Excessive Deferral	MAC defers for MAX_DEFERRAL: (two maximum size packets) = 0.24288 ms for 100Mbps. = 2.4288 ms for 10Mbps.
LostCrs	Lost Carrier Sense	Carrier sense is not detected or is dropped during the entire transmission of a packet (from the SFD to the CRC).
Tx10Stat	Transmit 10Mbps Status	= 1 if packet was transmitted via the 10Mbps interface. = 0 if packet was transmitted via MII.
LateColl	Late Collision	A collision occurs after 512 bit times (64 byte times).
TxPar	Transmit Parity Error	MAC transmit FIFO has detected a parity error.
Comp	Completion	MAC transmits or discards one packet.
TxHalted	Transmission Halted	Transmission was halted by clearing TxEn or setting TxHalt.
SQErr	Signal Quality Error	No heart beat signal observed at end of transmission.

Software reset to value 00_0000h. Cleared at beginning of each packet transmitted.

The transmission status flags are set whenever the corresponding event occurs. In addition, an interrupt is generated if the corresponding enable bit in the transmit control register is set.

The low order 5 bits can be read and masked as a single collision count, i.e when ExColl is 1, TxColl is 0. If TxColl is non-zero, then ExColl is 0.

Receive Control and Status Registers

Receive Control

(Rx_Ctl) 50h

15	14	13	12	11	10	9	8
-	EnGood	EnRxPar	-	EnLongErr	EnOver	EnCRCErr	EnAlign

7	6	5	4	3	2	1	0
-	IgnoreCRC	PassCtl	StripCRC	ShortEn	LongEn	RxHalt	RxEn

RxEn	Receive Enable	If zero, stop reception immediately.
RxHalt	Receive Halt Request	Halt reception after completing any current packet.
LongEn	Long Enable	Allow reception of frames longer than 1518 bytes. *
ShortEn	Short Enable	Allow reception of frames shorter than 64 bytes. *
StripCRC	Strip CRC Value	Check the CRC, but strip it from the message.
PassCtl	Pass Control Packets	Enable passing of received MAC Control packets to system.
IgnoreCRC	Ignore CRC Value	Do not check the CRC.

Interrupt Enable Flags

EnAlign	Enable Alignment	Interrupt upon receipt of a packet whose length in bits is not a multiple of eight, and whose CRC is invalid.
EnCRCErr	Enable CRC Error	Interrupt upon receipt of a packet whose CRC is invalid or, during its reception, the PHY asserts Rx_er.
EnOver	Enable Overflow	Interrupt upon receipt of a packet when the MAC receive FIFO is full.
EnLongErr	Enable Long Error	Interrupt upon receipt of a frame longer than 1518 bytes,* unless the long enable bit is set.
EnRxPar	Enable Receive Parity	Interrupt if the MAC receive FIFO detects a parity error.
EnGood	Enable Good	Interrupt upon receipt of a packet with no errors.

Hardware reset to value 0000h. Software reset clears RxEn and does not affect other values.

To receive an interrupt after each packet, set the good enable and all the error enable bits. Interrupts may also be enabled only for specific conditions.

* The frame lengths above do not include preamble and Start Frame Delimiter (SFD). See Chapter 6, page 69, in the section “MAC Frame and Packet Formats,” for more details.

Receive Status

(Rx_Stat) 54h

15	14	13	12	11	10	9
RxHalted	Good	RxPar	-	LongErr	Overflow	CRCErr

8	7	6	5	4	3	2	1	0
AlignErr	Rx10Stat	IntRx	CtlRecd	Reserved				

Receive Status Flags

CtlRecd	Control Received	Set if packet received is a MAC Control frame.
IntRx	Interrupt on Receive	Set if reception of packet caused an interrupt condition. This includes Good Received, if the EnGood bit is set.
Rx10Stat	Receive 10Mbits Status	= 1 if packet was received via the 10Mibits interface. = 0 if packet was received via MII.
AlignErr	Alignment Error	Frame length in bits was not a multiple of eight and the CRC was invalid.
CRCErr	CRC Error	CRC at end of packet did not match computed value, or the PHY asserted Rx_er during packet reception.
Overflow	Overflow Error	The MAC receive FIFO was full when it needed to store a received byte.
LongErr	Long Error	Received a frame longer than 1518 bytes.* Not set if the Long Enable bit in the receive control register is set.
RxPar	Receive Party Error	The MAC receive FIFO has detected a parity error.
Good	Good Received	Successfully received a packet with no errors. If EnGood = 1, an interrupt is generated on each packet received successfully.
RxHalted	Reception Halted	Reception interrupted by user clearing RxEn or setting RxHalt.

Software reset to value 0000h. Cleared at the beginning of each packet received.

* The frame lengths above do not include preamble and Start Frame Delimiter (SFD). See Chapter 6, page 69, in the section “MAC Frame and Packet Formats,” for more details.

The receive status flags are set whenever the corresponding event occurs. Once set, a flag stays set until another packet arrives. In addition, an interrupt is generated if the corresponding enable bit in the receive control register is set.

CtlRecd is set if the packet type is 8808h and the CAM recognizes the address. Software is responsible for checking the PAUSE operation code.

Software is responsible for separating alignment, CRC, and frame too long errors, and reporting them correctly as management information.

Station Management Data Access Registers

Station Management Data Register

(MD_Data) 58h

15	0
Station Management Data	

Software reset to value 0000h.

The MII section of the IEEE 802.3 Standard for 100-BASE-T, 100Mbps Ethernet, referenced in Chapter 1, “Reference Documents,” defines the format of the station management data registers. See specific PHY data sheets for additional hardware dependent registers.

Station Management Data Control and Address

(MD_CA) 5Ch

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PreSup	Busy	Wr	PHY					Addr				

Addr	Address	Address inside the PHY of register to read or write.
PHY	PHY Address	Address of PHY device to read or write.
Wr	Write	Set for write, clear for read.
Busy	Busy bit	Set to begin operation; controller clears when operation completes.
PreSup	Preamble Suppress	If set, the preamble is not sent to the PHY.

Software reset to value 0000h.

Before attempting to access the PHY Control registers, software should read the MD_CA register to ensure the busy bit is not set.

The contoller provides support for reading and writing of Station Management data to the PHY. Setting of options in station management registers does not affect the controller.

Some PHYs may not support the option to suppress preambles after the first operation.

CAM Access Registers

CAM Address

(CAM_Adr) 60h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CAM_Loc										0	0

CAM_Loc	CAM Address	The address of the four-byte CAM location to read or write.

Software reset to value 0000h.

In normal operation, the CAM_Adr and CAM data registers can read or write the CAM contents, including two double word locations immediately after the CAM for Flow Control Operation, as shown in Figure 6-3. Writing to other memory locations in normal operation has no effect. When the TestMode bit of the DMA Control register is set, the CAM_Adr can be used to read or write the entire DMA_RAM.

CAM Data

(CAM_Data) 64h

31	24	23	16	15	8	7	0
CAM_Data[0]		CAM_Data[1]		CAM_Data[2]		CAM_Data[3]	

Four bytes are accessed each time there is a read or write of the CAM Address register, i.e. the PCI Byte Enables are ignored. System software must take care to perform read, modify, and write logic when modifying only 2 bytes of a 4-byte word.

The CAM data register has a copy of the data stored at the CAM bytes addressed by the CAM address register. The register may be read more than once. When data is written to this register, the addressed CAM bytes are changed.

Note: Unlike data transferred in master mode, data transferred via this CAM interface is always interpreted internally as big endian.

CAM Enable

(CAM_Ena) 68h

Hardware reset to value 00_0000h. Software reset has no effect.

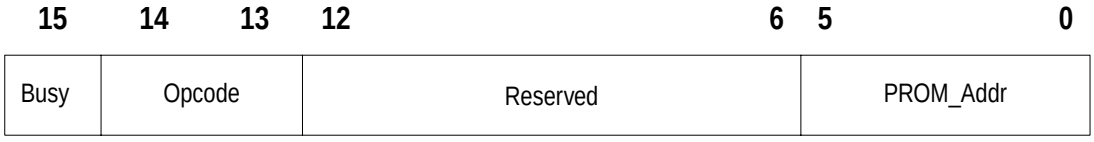
The CAM enable register indicates which entries are valid for address filtering. Up to 21 entries, numbered 0 through 20, may be active.

PROM Control Registers

The PROM control register provides control and status information and buffering for the PROM controller, which controls the reading and writing of an optional external EEPROM or small serial ROM device.

PROM Control

(PROM_Ctl) 6Ch



Busy	Busy bit	Set to begin operation. Will be cleared by the serial driver when operation is complete.
Opcode	Operation Code	1 0=Read 0 1=Write. 0 0=Enable or Disable Writing, as specified in PROM_Addr. [5:4] = 11, Enable [5:4] = 00, Disable 1 1=Erase
PROM_Addr	Address	Allows addressing of up to 64 16-bit entries.

Software reset to value 0000h.

Before attempting to access the EEPROM/ROM memory, software should read the PROM_Ctl register to ensure that the busy bit is not set.

PROM Data

(PROM_Data) 70h

15	0
PROM_Data	

Software reset to value 0000h.

The PROM Data registers provides the 16 bits of data written to or read from PROM.

The current implementation supports the following devices:

- MicroChip 93LC46B
- National NM93C46.

Upon hardware reset, the Subsystem Vendor ID Register, and Subsystem ID Register are loaded from the first two locations in the EEPROM or ROM. Software drivers are responsible for reading the station address, storing in the CAM, and enabling CAM operation, as part of driver intitialization.

When the PCI bus operates at 33MHz, the EEPROM or ROM is clocked at 1MHz.

System Error Count Registers

Missed Error Count

(Miss_Cnt) 7Ch

31	16	15	0
Reserved		Miss_Cnt	

Miss_Cnt	Missed Error Count	Counts the number of valid packets which are rejected by the MAC unit because the MAC receive FIFO overflows, a parity error occurs, or the Receive Enable bit (Rx_en) is cleared. This count excludes packets the CAM rejects.

Hardware reset to value 0000_0000h. Software reset has no effect. Cleared when read.

The missed error count register provides a count of packets discarded due to various types of errors. Together with status information for packets transmitted and received, this counter provides the information needed for station management.

Reading the missed error count register clears the register. It is then the responsibility of the system to maintain a global count with more bits of precision.

The missed error counter rolling over from 0x7FFF to 0x8000 sets the Missed

Roll bit in the MAC control register. It also generates an interrupt if the Enable Missed Roll bit is set.

If station management software wants more frequent interrupts, the missed error count register can be set to a value closer to the roll over value of 0x7FFF. For example, setting the register to 0x7F00 would provide for an interrupt after counting 256 occurrences.

Locations 0x74 and 0x78 are reversed to maintain software compatibility with earlier controller designs.

This chapter describes the data structures used by the PCI-based 100/10Mbps Ethernet Controller to communicate with the host system. These structures are located in system memory.

There are three types of basic data structures:

1. Frame Descriptors
2. Buffer Descriptors
3. Data Buffers

These data structures are used in three ways:

1. Transmit Queue - a list of frame descriptors for packets to transmit.
2. Receive Queue - a list of frame descriptors for packets that have been received.
3. Buffer List - a list of frame descriptors with unused buffers for receiving data.

The general organization of each of these was shown in Figure 1-4 on page 6. Each section of this chapter describes one of the basic data structures. Some data structures contain different information when they are used in different queues, which is described in sub-sections.

In continuous polling operation, a queue does not become empty once it is active. There is always a “dummy” frame descriptor at the end of the list, which belongs to the producer of new descriptors. For a detailed explanation, see Chapter 7, “DMA Operation.”

To begin transmission, the system stores into the transmit frame pointer register the address of the first frame descriptor in the transmit queue. The controller traverses the transmit queue, updating the status of transmitted packets. Transmission complete is indicated in the frame descriptor status field and in the ownership bit of the frame descriptor control field. This allows the queue to be processed by system software after transmission, e.g. to free buffers.

The controller acquires buffers from the buffer list, and writes new frame descriptors and new buffer descriptors into the free descriptor area, as described in Chapter 7, “DMA Operation.”

Frame Descriptors

Each frame descriptor has a pointer to the next frame descriptor in the queue, a system data field, a length field, and control and status fields. Table 5-1 shows the layout of a frame descriptor.

Table 5-1 Frame Descriptor Format

Byte 3	Byte 2	Byte 1	Byte 0	Offset
FDNext				00
FDSystem				04
Reserved	FDStat			08
FDCtl		FDLength		0C

FDNext	Next Frame Descriptor	Address of next frame descriptor in this queue.
FDSystem	Frame System Data	For use by the system or application software.
FDStat	Frame Descriptor Status	Status field for this Frame Descriptor.
FDCtl	Frame Descriptor Control	Control field for this Frame Descriptor.
FDLength	Frame Length	Length field for this Frame.

The controller will preserve the contents of the frame system data field, FDSystem. This field may be used by either the system or the application programs. The initial value for frame descriptors written on the receive queue is obtained from the current buffer list frame descriptor.

Each queue makes slightly different use of the FDNext, FDCtl, FDStat, and FDLength fields, as explained below.

Next Frame Descriptor Field (FDNext)

The next frame descriptor field containers either an End-Of-List (EOL) flag, or a pointer to the next frame descriptor in the same queue. Frame descriptors must be aligned to 16-byte boundaries, i.e. a valid pointer must have bits 0-3 set to zero.

31 … 4	3	2	1	0
Pointer	0	0	0	EOL

Pointer	28-bit Pointer Field	If EOL=0, contains upper 28 bits of address of the next frame descriptor in this queue.
EOL	End-Of-List Flag	=0. Pointer is valid. =1. End of list. Must wait for flag to clear.

On all the queues, the next frame descriptor field is used to stop the consumer of the list, by setting the EOL bit. The consumer must wait for the producer of the list to clear the EOL bit, when it stores a valid pointer. On the buffer list queue, it is possible to chain from one buffer pool to another using the next frame descriptor field. If

chaining of buffer lists is not used, the software drivers should set the FDNext field to contain its own address. This would cause the controller to re-examine the same buffer area for re-use. Alternatively, the EOL bit can be set, causing the controller to stop.

Frame Descriptor System Field (FDSystem)

The FDSystem field is a 32-bit field which is reserved for system software use. It could be a pointer to a table of information, a pointer to C++ virtual functions, etc.

On the transmit queue, the FDSystem field is not used.

On the receive queue, the controller will copy the contents of the FDSystem field from the current buffer list queue, where the first buffer descriptor was allocated.

Frame Descriptor Status Field (FDStat)

On the transmit and receive queues, the FDStat field is used for reporting transmission and reception completion status. For a description of the status bits, see the Tx_Stat and Rx_Stat registers in Chapter 4, page 47 in section “MAC Layer Registers.”

On the receive buffer list, the FDStat field is not used.

Frame Descriptor Length Field (FDLength)

On the transmit queue, the FDLength field is not used.

On the receive queue, the controller sets the FDLength field to the total length of the packet.

On the buffer list queue, the FDLength field is used to count the number of free buffer descriptors allocated to the queue. The controller accesses the buffer list frame descriptor via the buffer list frame pointer register. If the controller encounters a buffer it does not own, it will set the BL_Ex bit in the Interrupt Source register and wait for the system to clear it. The controller will read the buffer descriptors, using the FDLength field as a limit. When the controller nears the end of the list, it will fetch the next frame descriptor pointed to by the FDNext field as described in Chapter 4, page 45 in section “Descriptor Area Registers.”

Frame Descriptor Control Field (FDCtl)

The table below shows the abbreviation, field name, description, and usage of the FDCtl field.

15	14 13 12 11 10	9 8 7 6 5 4 3 2 1 0
COwnsFD	FrmOpt	Reserved	BDCount

Symbol	Name	Description	Used by
COwnsFD	Controller Owns Frame Descriptor	=1. The controller owns the frame descriptor, after the system sets COwnsFD. =0. The system owns the frame descriptor, after the controller clears COwnsFD.	Tx, Rx
FrmOpt	Frame Options	Per Frame Control Options. See below.	Tx,BL
BDCount	Buffer Descriptor Count	Number of BD's allocated. (0 to 29)	Tx,Rx

The ownership field is used in the transmit and receive queues to synchronize processing by the controller and the system. Frame options are used in the transmit queue and the buffer list. The buffer descriptor count field is only used in the transmit and recieve queues. (The buffer list uses the length field as a count field to allow larger buffer pools). If an attempt is made to use more than 28 buffer descriptors for a single received packet, then an Excess Buffer Descriptor error is generated.

The transmit queue uses the frame options field to set transmit characteristics for individual packets:

- 10000 = Big Endian byte ordering.
- 01000 = Interrupt after transmitting.
- 00100 = No CRC appended.
- 00010 = No PAD bytes, if short frame.

These bits can be combined to ask for combinations of characteristics. For example, 01110 would mean: little endian, interrupt after transmission, do not append CRC, and do not pad a short packet.

Per packet Big Endian controls can be useful in a hub application, where packets are received for transmission from a mixture of Big and Little Endian sources. For computer applications, it is easier to use the global Big Endian control bit described in Chapter 4, page 43, in the section "DMA Control."

The receive buffer list queue uses the frame options field to control packing and Little or Big Endian data order, providing that the global enable bit described in Chapter 4, page 46 in section "Receive Fragment Size Register," has not been set:

- 10000 = Big Endian byte ordering.
- 00001 = Enable buffer packing for the buffers in this frame, ignore the global enable bit.

If packing is enabled, the RxFragSize register controls the the packing algorithm.

Buffer Descriptors

Each buffer descriptor has a pointer to the data buffer, control and status bytes, and a two-byte length field. Table 5-2 shows the layout of a buffer descriptor.

Table 5-2 Buffer Descriptor Format

Byte 3	Byte 2	Byte 1	Byte 0	Offset
BuffData				00
BDCtl	BDStat	BuffLength		04

BuffData	Buffer Data Pointer	32-bit address of storage for bytes of data.
BDCtl	Buffer Descriptor Control	Control for this buffer descriptor.
BDStat	Buffer Descriptor Status	Status for this buffer descriptor.
BuffLength	Buffer Length	Length field for this buffer descriptor.

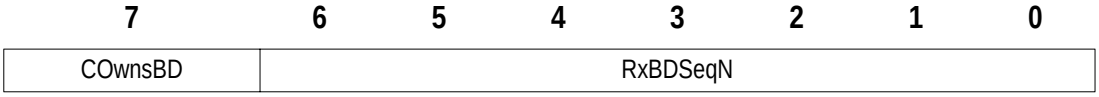
When buffers are on the buffer list queue, the BuffData field points to the beginning of the buffer, and the BuffLength field reflects the allocated size of the unused buffer. When buffers are in use on the transmit and receive queues, the BuffData field points to the beginning of data, and the BuffLength field reflects the length of the data.

It is the responsibility of the system software to set the length field to the allocated size, when buffers are placed in the buffer list queue.

As with frame descriptors, each queue makes slightly different use of the BDCtl and BDStat fields, as explained below.

Buffer Descriptor Control (BDCtl)

On the transmit queue, the BDCtl field is not used. On the receive queue, the BDCtl field holds the number of buffer descriptors pointing into a single buffer area. The first buffer descriptor in the frame is numbered 0, the next 1, etc.



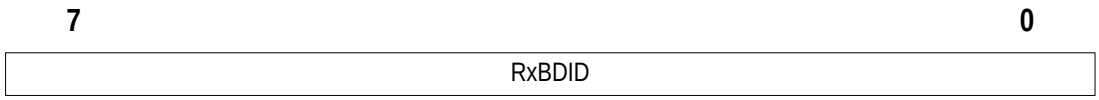
COwnsBD	Controller Owns Buffer Descriptor	=1. Controller owns the buffer descriptor. When the system sets COwnsBD, the buffer is free for reception. =0. System owns the buffer descriptor. When the controller clears COwnsBD, the buffer has been filled.
RxBDSeqN	Receive Buffer Descriptor Sequence Number	After reception: the sequence number for this buffer within the current buffer area.

On the free buffer list, the BDCtl field is used to record ownership of the buffer. This allows synchronizing allocation and freeing of buffer descriptors, to ensure that the controller will not “wrap around” and begin reusing buffers before the system can empty them.

Buffer Descriptor Status Field (BDStat)

On the transmit queue, the BDStat field is not used by the controller.

On the receive queue, the BDStat field is used as a Buffer ID, which is copied from the free buffer queue.



RxBDID	Receive Buffer Descriptor ID	The buffer ID value.

On the free buffer queue the BDStat field is used to pass the buffer descriptor ID number to the controller.

Note: Buffer IDs can only be unique if there are at most 256 buffers in a single buffer pool.

This chapter gives detailed information about the following aspects of MAC operation:

1. MAC frame and packet formats
2. Destination address formats
3. Initialization
4. MAC Register access
5. Transmitting a frame
6. Receiving a frame
7. CAM operation
8. Full Duplex PAUSE Operation
9. Error signaling and network management
10. Accessing station management data
11. Accessing an EEPROM or ROM.

MAC Frame and Packet Formats

Table 6-1 shows the format of an IEEE 802.3/Ethernet frame. The standard packet has the following fields:

Table 6-1 Fields of an IEEE 802.3/Ethernet packet (frame)

Packet (encoded on the medium)							
Added by transmitter, stripped by receiver		Data Frame (sent by user)					Added by transmitter
		Data Frame (delivered to user)				Optionally stripped by receiver	
Preamble	SFD	Destination address	Source address	Length or Type	LLC data	Pad bytes	Cyclic Redundancy Check(CRC)
7 bytes	1	6 bytes	6 bytes	2	0-1500	0-46	4 bytes
				high low	...	. . .	msb lsb

1. Preamble–seven identical bytes. The bits in each byte are 10101010, transmitted from left to right.

2. Start Frame Delimiter (SFD)–one byte. The bits are 10101011, transmitted from left to right.

3. Destination Address–six bytes. May be an individual or a multicast (including broadcast) address.

4. Source Address–six bytes. MAC does not interpret the bytes. However, to be a valid station address, the first bit transmitted, the LSB of the first byte, should be set to 0.

5. Length or Type–two bytes. The MAC only recognizes the special value of 8808h as the MAC Control Frame type. Other values are not processed. The current IEEE 802.3 Standard specifies that values less than 1500 are lengths and values greater that 1535 are types. Values less that or equal to 1500 indicate the number of the Logical Link Control (LLC) data bytes in the data field, in bytes. The MAC transmits the high-order byte of the Length or Type Field first.

6. Logical Link Control (LLC) Data–46 to 1500 bytes (including Pad).

7. Pad–0 to 46 bytes. If the LLC Data is less than 46 bytes long, the MAC will transmit pad bytes of zeroes.

8. Cyclic Redundancy Check (CRC)–four bytes. A value computed as a function of all fields except the preamble, the SFD, and the CRC itself. The IEEE 802.3 standard also refers to the CRC as the Frame Check Sequence (FCS).

The Preamble, SFD, Pad Data, and CRC are added by the transmitter. Padding can also be done in software, and there is a Transmit Control bit to suppress CRC addition. The Receive Control register has a bit to control stripping the CRC. Stripping of Pad Data is the responsibility of the software drivers.

The MAC transmits the least significant bit of each byte first for all fields except the CRC. Throughout this document, we attempt to use “packet” to denote all of the bytes transmitted and received, while “frame” refers to the bytes delivered by the user for transmission, and to the user who is receiving.

There are a number of factors and options which can affect this “standard MAC frame”:

- Some PHYs may deliver a longer or shorter preamble.
- Short packet mode allows LLC data fields with less than 46 bytes. There are options to suppress padding and allow reception of short packets.
- Long packet mode allows LLC data fields with more than 1500 bytes. There is an option to allow reception of long packets.
- No CRC mode suppresses the appending of an CRC field.
- Ignore CRC mode allows the reception of packets without valid CRC fields.
- Pad bytes are counted in the length field as part of the LLC data.

Destination Address Format

Bit 0 of the destination address is an address type designation bit. It identifies the address as either an individual or a group address. Group addresses are also called multicast addresses. Individual addresses are also called unicast addresses. The broadcast address is a special group address, namely ff-ff-ff-ff-ff-ff in hex. There is a special group address used in connection with the Full Duplex PAUSE Operation: 01-80-C2-01-00-01.

Bit 1 distinguishes between locally or globally administered addresses. For globally administered (or U, universal) addresses, the bit is set to 0. If an address is to be assigned locally, this bit is set to 1. For the broadcast address, this bit is also a 1.

Destination address, first byte:

7	6	5	4	3	2	1	0
Rest						U/L	I/G

I/G	Individual or Group Flag	=0 Individual address. =1 Group address.
U/L	Universal or Local Flag	=0 Universal address. =1 Local address.
Rest	Rest of Byte	Rest of first byte of destination address.

Special Flow Control Destination Address

The current specification for Full Duplex Flow Control specifies a special destination address for the PAUSE Operation packet. In order for the MAC to receive packets which contain this special destination address, the address must be programmed into one of the CAM entries, that CAM entry must be enabled, and the CAM must be activated.

Some CAM entries are used when generating a Flow Control packet via the SdPause bit in the Transmit Control register, as explained later in this chapter.

Initialization

On power up and reset, the MAC control and status registers are set as described in Chapter 4, “Registers.” Refer to the verification test suite and software application notes for more details about order of initialization, and functional dependencies.

Transmit Collision Count, CAM Data, and EEPROM/ROM Buffer registers are not set on power up or reset. The Transmit Collision Count register is reset at the beginning of transmitting a new packet. The CAM memory should be initialized before enabling usage of the CAM.

MAC Register Access

MAC Register access is controlled by the PCI bus interface. For more information on register access, including MAC command and status registers, see Chapter 7, “DMA Operation.”

Special Register Clear Operations

The missed packet error count register is cleared on read. This ensures synchronization with software drivers which accumulate a total count.

The transmit and receive status registers are cleared at the beginning of the next packet. For this reason, the values read from the register interface may not be stable. The value of these registers is stored in the FDStatus field of the Frame Descriptor in memory, for each packet transmitted or received, and software should examine the status values found in system data structures.

Transmitting a Frame

1. To transmit a frame, the transmit enable bit in the transmit control register must be set and the transmit halt request bit must be zero. In addition, the halt immediate and halt request bits in the MAC control register must be cleared. These conditions are normally set after DMA controller initialization has occurred, such as storing a valid frame descriptor address in the transmit frame pointer register. The MAC will then signal the DMA engine to transfer bytes to the MAC transmit FIFO. The DMA transmit controller then controls the transfer of bytes to the MAC transmit FIFO.

2. The MAC transmit block will start transmitting the data in the FIFO, but will retain the first 64 bytes until it has acquired the net. At that time, the MAC transmit block will request more data and transmit it until the DMA transmit controller signals the end of data to be transmitted. The MAC transmit block generates pad bytes, if needed, appends the calculated CRC to the end of the packet, and transmission ends. It sets the completion bit in the transmit status register, signaling the end of a transmission, which may in turn cause an interrupt.

3. Transmission of data across the MII interface is driven by either a 25 or 2.5MHz MII transmit clock, Tx_clk. Transmission across the 10Mbits endec is driven by a 10MHz transmit clock, TxC_10.

4. The MAC transmit block does not begin transmission onto the net until there are eight bytes of data in the MAC transmit FIFO. Since the first eight bytes transmitted are the preamble and the Start Frame Delimiter, this gives an initial 16 byte times for DMA latency. The DMA transmit block does not begin transferring data to the MAC transmit FIFO, until either the entire

packet is in the DMA RAM buffer, or the number of bytes in the DMA RAM buffer exceeds the DMA transmit threshold register. If transmit underrun errors occur, the problem can be corrected by setting the DMA transmit threshold register to a higher value.

5. The MAC transmit block will check the parity. If there is a parity error, the MAC transmit block resets the FIFO, and sets the MAC parity error bit in the transmit status register.

The IEEE 802.3 CSMA/CD MAC-layer Protocols

- The MAC transmit block consists of three state machines. The main transmit state machine implements the MAC-layer protocols, and controls the other two. The gap state machine tracks and counts the inter-packet gap timing between packets. And the back-off state machine implements the backoff and retry algorithm of the 802.3 CSMA/CD protocol.
- In half-duplex mode, the gap state machine is responsible for counting the 96 bit times from the deassertion of the carrier sense signal, which is the inter-record gap. It breaks the 96 bit times for the inter-record gap into the first 64 bits, and the last 32 bits, in order to precisely control the appropriate times for beginning transmission. If there is any traffic within the first 64-bit times, it resets the counter and resumes counting from zero. If there is any traffic within the last 32 bits, it continues counting and signals the end at 96 bit times.
- In full-duplex mode, the gap state machine starts counting at the end of transmission and signals the end at 96 bit times.
- If the main transmit state machine detects a collision, it starts the back-off state machine counters and waits for the end of the back-off slot, before retransmitting the collision-causing packet again. Each time there is a collision for the same packet, the back-off state machine increments an internal attempt counter. An 11-bit pseudo-random-number generator outputs a random number by selecting a subset of the value of the generator. The subset grows by one bit for each subsequent attempt. This implements the equation:

$$0 \leq r < 2^{k}$$
$$k = \min(n, 10)$$

  where r is the number of slot times that the MAC has to wait in case of a collision, and n is the number of attempts. For example, after the first collision, n is 1 and r is a random number between 0 and 1. The pseudo-random-number generator in this case is one-bit wide and gives a random number of either 0 or 1. After the second attempt, r is a random number between 0 and 3; the state machine looks at the two least significant bits of the generator ($n = 2$) which gives a value between 0 and 3.
- In order to improve the statistical independence between two MACs using the same pseudo-random number generator, the MAC uses values from the CRC of previous successfully transmitted packets to modify the basic random number sequence.

The MII Transmit Operation

- If there is data to be transferred, the inter-packet gap is OK, and the MII is ready (there are no collisions, and either in full-duplex mode or there is no CrS), then the MAC transmit block transmits the preamble followed by the SFD. After the transmission of the preamble and the SFD, it transmits 64 bytes of data regardless of the packet length, unless short transmission is enabled. This means that if the packet is less than 64 bytes, it will pad the LLC data field with zeroes. At the end of the packet, it appends the CRC, if CRC generation is enabled. If there is any collision during this first 64 bytes (8 bytes of preamble and SFD and 56 bytes of the frame), it stops the transmission and transmits a jam pattern (32 bits of all ones). It increments the collision attempt counter, returns control to the back-off state machine , and retransmits the packet when the backoff time has elapsed and the gap time is OK.
- If there are no collisions, the MAC transmit block transmits the rest of the packet, and at this time (after the first 64 bytes have been transmitted without collisions), it allows the DMA engine to overwrite this packet. After transmitting the first 64 bytes, it transmits the rest of the packet and appends the CRC to the end. FIFO underrun or more than 16 collisions will cause the state machine to abort the packet (no retry) and prepare for the next packet in the queue.
- In case of any transmission errors, the MAC transmit block sets the appropriate error bit in the transmit status register, and it may generate an interrupt, depending on the transmit control register.
- Figures 6-1 and 6-2 show the timing relationships among MII signals.

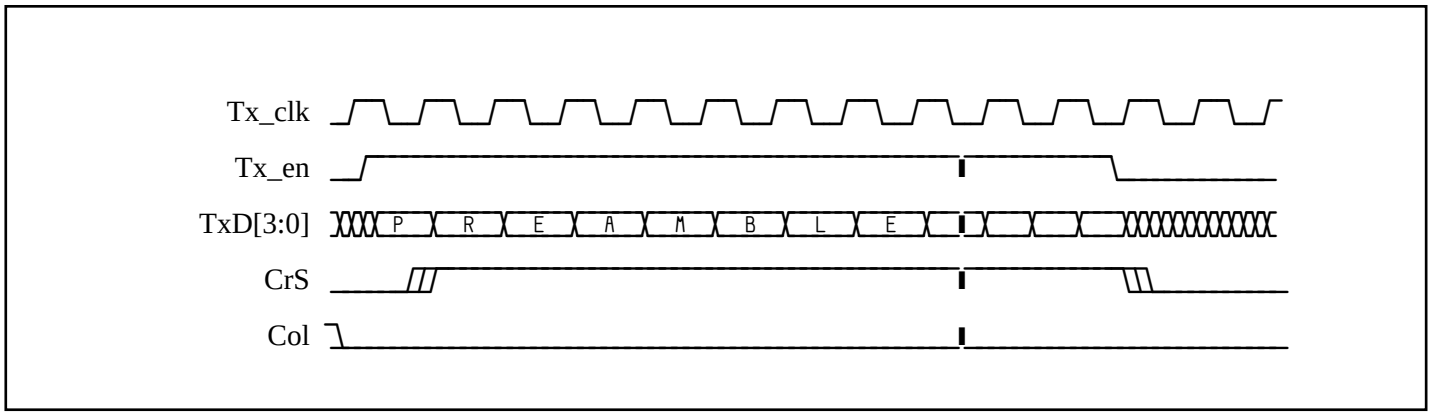


Figure 6-1 Transmission Without Collision

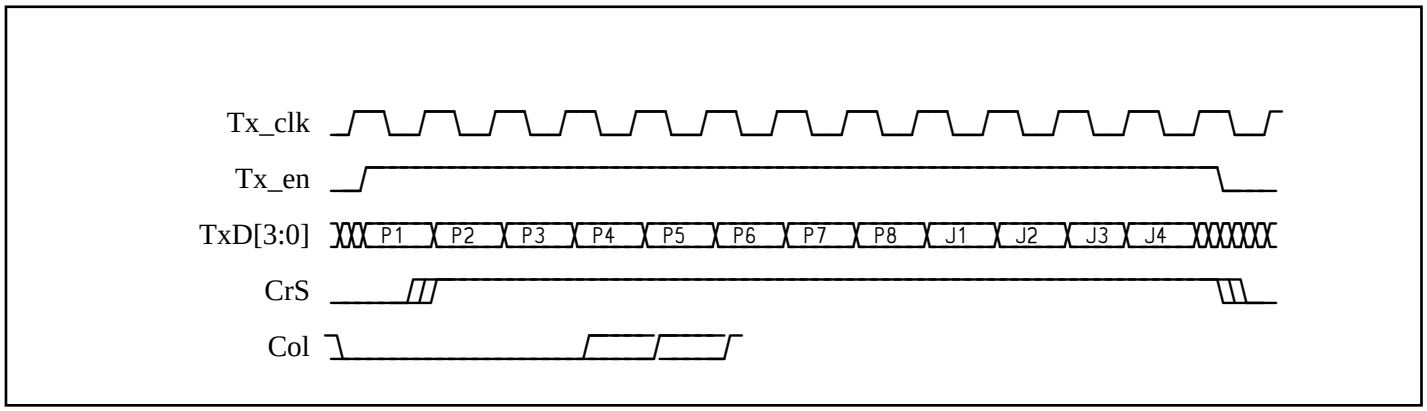


Figure 6-2 Transmission With Collision in Preamble

Receiving a Frame

1. To receive a frame, the receive enable bit in the receive control register must be set and the receive halt request bit must be zero. In addition, the halt immediate and halt request bits in the MAC control register must be cleared. These conditions are normally set after DMA controller initialization has occurred, such as storing a valid address into the buffer list frame pointer register, and initializing the free descriptor area base and registers. The DMA receive controller then controls the transfer of bytes from the MAC receive FIFO. For more information on initializing the data structures to enable reception, see section “Receiving a Frame” in Chapter 7.

2. The MAC receive block, when enabled, constantly monitors a data stream coming from either the MII or the optional external 10Mbits endec. If the MAC is in loopback mode, the data stream will be coming from the MAC transmit block via the MII or 10Mbits endec lines.

3. The MAC receive block receives zero to seven bytes of preamble, followed by the Start Frame Delimiter (SFD). The MAC receive block waits for the SFD pattern before receiving the packet.

4. The first nibble of destination address follows the SFD. When it has received a byte, the MAC receive block generates parity, stores the byte with its parity in the MAC receive FIFO, and signals that data is present. It combines subsequent nibbles into bytes and stores them in the FIFO. The DMA receive controller reads bytes from the MAC receive FIFO, checks parity, and moves the data into the DMA receive FIFO. When the MAC receive FIFO becomes empty, or when it drives out the last byte of a packet, the MAC receive block signals these conditions.

5. If, during frame reception, the PHY asserts both Rx_DV and Rx_er, the MAC receive block reports a CRC error for the current packet.

6. After the MAC receive block receives the destination address, the CAM block attempts to recognize it. If the CAM block rejects the packet, the MAC receive block signals this condition and the DMA receive block discards the data packet.

CAM Operation

To read or write the CAM, system software should first set the CAM address register, then read or write the CAM data register. All bytes are written, without regard to partial word enables. When writing the upper or lower two bytes of a double word, it is the responsibility of the driver software to correctly write the adjacent two byte field, as well. The controller does not support read/modify/write cycles to its internal DMA RAM.

Figure 6-3 shows how the MAC reads CAM entries from the CAM memory. Entries are assumed to be in Big endian order: #0-0 is the first byte of the first entry, #0-5 is the sixth and last byte of the first entry, and so on. There are two bytes after CAM entry #20, Rsv-2 and Rsv-3, and two double words. MC#1 and MC#2, which are not used in CAM operation, but are used in generating MAC Control Frames, as explained in the next section entitled “Full Duplex PAUSE Operation.”

Byte 3	Byte 2	Byte 1	Byte 0	
#0-0	#0-1	#0-2	#0-3	00
#0-4	#0-5	#1-0	#1-1	04
#1-2	#1-3	#1-4	#1-5	08
#2-0	#2-1	#2-2	#2-3	0C
#2-4	#2-5	#3-0	#3-1	10
#3-2	#3-3	#3-4	#3-5	14
#4-0	#4-1	#4-2	#4-3	18
#4-4	#4-5	#5-0	#5-1	1C
#5-2	#5-3	#5-4	#5-5	20
				...
#18-0	#18-1	#18-2	#18-3	6C
#18-4	#18-5	#19-0	#19-1	70
#19-2	#19-3	#19-4	#19-5	74
#20-0	#20-1	#20-2	#20-3	78
#20-4	#20-5	Rsv-2	Rsv-3	7C
MC#1-0	MC#1-1	MC#1-2	MC#1-3	80
MC#2-0	MC#2-1	MC#2-2	MC#2-3	84

Figure 6-3 CAM Memory Map

Full Duplex PAUSE Operation

Transmit Pause Operation

To enable Full Duplex PAUSE Operation, the Special Broadcast address for MAC Control Packets must be programmed into the CAM, and the corresponding CAM Enable bit set. While this can be any CAM location, the next section will specify how some CAM locations may be preferred, to optimize CAM entry utilization.

The MAC Receive circuit recognizes the Full Duplex PAUSE Operation when the following conditions are met:

- The Type/Length field has the Special value for MAC Control packets, 0x8808.
- The packet is recognized by the CAM.
- The length of the packet is 64 bytes.
- The operation field specifies PAUSE operation.

When a Full Duplex PAUSE Operation is recognized, the MAC Receive circuit loads the operand value into the Pause Count Register, and signals both the MAC DMA engine that pause should begin at the end of the current packet, if any.

The Pause circuit maintains the Pause Counter, and decrements it to zero, before signalling the end of the pause operation, and allowing the DMA Transmit circuit to resume.

If a second Full Duplex PAUSE Operation is recognized while the first operation is in effect, the Pause Counter is reset with the current operand value. Note that a value of 0 may cause premature termination of a pause operation in progress.

Remote Pause Operation

The Toshiba TC35815CF Flow Control 10/100Mbps Ethernet Stand-alone MAC supports full programmability of the MAC Control frames to support both PAUSE Operation and future uses of MAC Control.

To send a remote PAUSE Operation or other MAC Control Frame the following steps need to be taken:

- Program CAM location #0 with the Destination Address.
- Program CAM location #1 with the Source Address.
- Program CAM location #20 with the MAC Control type Field, PAUSE Operation opcode, and operand value. The two reserved bytes after CAM location #20 should be written with 0000h.
- Program the two double word locations MC#1 and MC#2, with 0000_0000h.
- Write the Transmit Control Register, setting the SdPause bit.

The Destination Address and Source Address are normally the Special Broadcast Address for MAC Control Frames and the local Station Address, respectively. These CAM entries can be enabled for use in address filtering. CAM entry #20 should not be enabled, when used as part of Flow Control transmission.

Upon completion, the transmit status is written to the Transmit Control Frame Status register. The DMA engine generates an interrupt if the Transmit Control Complete Enable bit (10) of the Interrupt Enable control register is set.

Error Signaling

The error and abnormal operation flags set by the MAC are arranged into transmit and receive groups, and can be found in either the transmit status register (Tx_stat) or the receive status register (Rx_stat). In addition, the missed packet register counts packets missed for system network management purposes. Please refer to Chapter 4, “Registers,” for the formats of the flags and counters.

Reporting of Errors in Transmit

Transmit operation terminates when the entire packet (preamble, SFD, data, and CRC) has been successfully transmitted to the physical medium without encountering a collision. In addition, the MAC transmit block detects and reports both internal and network errors.

Under the following conditions, transmission will be aborted and a status bit will be set. Many of the bits that are set can also generate interrupts, if the corresponding interrupt enable bit has been set in the transmit control register.

MAC Transmit Parity Error	A parity bit protects data coming from the DMA transmit controller via the DII into the MAC transmit FIFO. A parity error sets the TxParErr bit of the transmit status register and generates an interrupt, if interrupt is enabled.
MAC Transmit FIFO Underrun	The 80-byte MAC transmit FIFO is capable of handling a worst case DMA latency of 1.28μs (128 bit times, or 16 byte times), because 64 bytes are retained for possible retransmission after a collision. A MAC transmit FIFO underrun indicates a PCI bus latency problem, since the DMA transmit controller has more than enough bandwidth to keep up. Such an underrun sets the underrun bit in the transmit status register.
Lost Carrier	Carrier Sense (CrS) is monitored from the beginning of the Start Frame Delimiter (SFD) to the last byte transmitted. A lost-carrier condition indicates that CrS was never present or was dropped during transmission (a possible network problem), but transmission is **not** aborted. During loopback mode, Tx_en drives CrS. During full-duplex operation, CrS is not passed to the transmit block, and lost carrier will not be asserted. Lost carrier sets the LostCrS bit in the transmit status register.
Excessive Collision	Whenever the MAC encounters a collision during transmit, it will back off, update the collision counter, and try again later. When the counter equals 16 (16 attempts all resulted in a collision) transmission is aborted. Excessive collisions probably indicate a network problem. Excessive collision sets the ExColl bit in the transmit status register.
Late Collision (Transmit Out-Of-Window Collision)	In a correctly operating network, the controller sees a collision (if there is one) within the first 64 bytes of data being transmitted. If a collision occurs after this time a possible network problem is detected. Late collision sets the LateColl bit in the transmit status register, and transmission of the packet is aborted.
Signal Quality Error (SQE)	In 10 Mb/s mode, the MAC checks for a “heartbeat” at the end of a transmitted packet. This is a short Collision signal within the first 40 bit times after end of transmission. Signal Quality Error sets the SQErr bit in the Transmit Status register.
Deferral	During an attempt to send a packet, the MAC may have to defer the transmission because of the pre-occupied network. This is not an error, but is used as a network activity indicator, but only when collisions do not occur. Deferral sets the TxDeferred bit of the transmit status register.
Excessive Deferral	During the first attempt of sending a packet, the MAC may have to defer the transmission because of pre-occupied network. If the deferral time is longer than two maximum sized packets (2.4288ms for either of the 10Mbits operation modes or 0.24288ms for the 100Mbits operation mode) transmission is aborted if excessive deferral is enabled. Excessive deferral indicates a possible network problem. Excessive deferral sets the ExDefer bit of the transmit status register.
Paused	During an attempt to send a packet, the MAC may have to defer the transmission because the Transmitter has been paused by the reception of MAC Control packet continuing a PAUSE Operation. This is not an error, but is used as a network activity indicator. To assist software in marking packets which experience a pause, the Paused bit is set on the last packet before a PAUSE will take effect.

Reporting of Errors in Receive

The MAC receive block starts putting received data from the physical medium into the MAC receive FIFO after detecting the Start Frame Delimiter (SFD). It also checks for MAC receive FIFO overflow during reception. At the end of reception, the MAC receive block looks for external errors (Alignment, Length, CRC, and Frame Too Long).

MAC Receive Parity Error	A parity bit protects data once it enters the MAC receive FIFO. A parity error sets the RxParErr bit of the receive status register and generates an interrupt, if interrupt is enabled.
Alignment Error	At the end of reception, the MAC receive block checks that the incoming packet has been correctly framed on an 8-bit boundary. If it is not, and the CRC is invalid, data has been disrupted through the network, and the MAC receive block reports an alignment error. A CRC error is also reported. The AlignErr bit and the CRCErr bits are set in the receive status register.
CRC Error	At the end of reception, the MAC receive block checks the CRC for validity, and reports a CRC error if it is invalid. CRC, Frame Alignment, and Long errors are the network errors detected by the receive unit. They might be detected in the following combinations. – CRC error only. – Frame Alignment and CRC errors only. – Long and CRC errors only. – Frame Alignment, Long, and CRC errors.
Overflow Error	During reception, the incoming data are put into the MAC receive FIFO before they are transferred to the DMA receive controller. If the MAC receive FIFO fills up because of excessive system latency or other reasons, the MAC receive block rejects the received packet and sets the Overflow Error bit of the receive status register.
Long Error	The MAC receive block checks the length of the incoming packet at the end of reception. If the length is longer than the maximum frame size of 1518 bytes, the MAC receive block reports receiving a Long Error, unless long frame mode is enabled.
MII Error	The PHY informs the MAC if it detects a media error (such as coding violation) by asserting Rx_er. When the MAC sees Rx_er asserted, a CRC Error is forced at the end of the packet.

Accessing Station Management Data

The basic sequence of events in accessing the station management data are:

- System software reads the busy bit to ensure the MD is not busy.
- For a write operation, the data should be written into the data register before setting the control register.
- Software writes the MDC address, the read/write flag, and sets the busy bit.
- The controller completes the operation, and clears the busy bit.
- For a read operation, when system software detects the busy bit is cleared, it can read the data register.

Accessing an EEPROM or ROM

The basic sequence of events in accessing an external EEPROM or serial ROM are:

- System software reads the busy bit to ensure the driver is not busy.
- For a write operation, the data should be written into the data register before setting the control register.
- Software writes the address and the read/write flag, and sets the busy bit.
- The controller completes the operation, and clears the busy bit
- For a read operation, when system software detects the busy bit is cleared, it can read the data register.

This page left blank intentionally

This chapter gives information on programming the TC35815CF Flow Control 10/100Mbps Ethernet Controller. Some topics will be covered later in application notes. Programming details differ slightly, depending on the control mode chosen: batch processing or continuous polling. The descriptions in this chapter assume continuous polling, unless otherwise noted.

The topics covered in this chapter include:

1. PCI Initialization
2. DMA and MAC Initialization
3. Queue Initialization
4. Transmitting a Frame
5. Receiving a Frame
6. Processing Interrupts

PCI Initialization

At system initialization time, the IDSel signal can be used to write the PCI configuration registers. This is normally done by having the system map the controller into a startup memory address space, and having data transfers to those addresses generate the IDSEL signal. Alternatively, some registers can be initialized from an optional EEPROM or ROM, as described in the Chapter 6 “Accessing an EEPROM or ROM” section.

The registers that must be initialized include:

- PCI I/O or Memory Base Address - to map registers into I/O or memory space.
- PCI Command - to customize PCI capabilities.

The registers that might be initialized include:

- PCI Interrupt - to customize latency or route interrupts.

DMA and MAC Initialization

After PCI initialiazation, the DMA and MAC control registers are normally mapped into I/O space, or memory space, and can be read or written from the mapped space. Alternatively, these registers can be initialized from an optional EEPROM or ROM, as described in Chapter 6, page 81, in the section “Accessing an EEPROM or ROM”.

The registers that must be initialized include:

- DMA Transmit Frame Pointer - to initiate transmission.
- DMA Buffer List Frame Pointer - to provide buffers for reception.
- DMA Free Descriptor Area Base and Limit - to initialize receive notification area.
- DMA Transmit Polling Count - to customize polling for packets to transmit.
- DMA Transmit Threshold - to customize handling of transmit latency.
- MAC Transmit Control - to change default transmission settings.
- MAC Receive Control - to change default reception settings.
- MAC CAM Control - to customize station and multicast-group address recognition.
- MAC CAM Address and Data - to provide station address and other address filtering.
- MAC CAM Enable - to enable individual CAM entries after setup.

The registers that might be initialized include:

- MAC Control Register - to customize MAC configuration.
- DMA Transmit Burst Size - to customize transfer sizes.

Queue Initialization

Before starting the controller, the system needs to set up the transmit queue, the buffer list queue, and the receive descriptor area.

Transmit Queue Initialization

There are two modes of operation for the transmitter: batch processing and continuous polling. For batch processing, the system software sets up a linked list of frame descriptors to transmit, with the last frame descriptor containing an End-of-List (EOL) indicator. When the last frame descriptor is transmitted, the transmit frame pointer register loads the EOL indicator, and transmission terminates. Later, the system must restart transmission by storing a new value in this register.

For continuous polling, the system software sets up a linked list of frame descriptors to transmit, which ends with a dummy frame descriptor. The linked list may be initially empty, except for the dummy frame descriptor. The dummy frame descriptor is owned by the system, to prevent the controller from accessing it. When a new packet is to be transmitted, the dummy frame descriptor is overwritten, as described in this chapter “Transmitting a Frame”.

Buffer List Initialization

The buffer list queue is initialized by setting up a linked list of frame descriptors with one or more frame descriptors, each containing a list of free buffer descriptors. The list can be any one of the following:

1. A single frame descriptor, with a large number of free buffer descriptors.

2. A linked list of frame descriptors.

3. A circular queue, with the last frame descriptor pointing to the first frame descriptor.

For 1.) and 2.) the FDNext field would have the EOL bit set; for 3.) the FDNext field of the last frame descriptor would point to the first frame descriptor. The receive buffer fragment size register can be set to globally enable packed buffer usage. Alternatively, the frame descriptor control field (FDCtl) can be used to select packed or unpacked buffer usage on a per buffer area basis. For packed buffer usage, the buffer ID fields can be set, to assist in memory management. For more details on enabling buffer packing see Chapter 4, page 46 in the section “Receive Fragment Size Register”, Chapter 4, page 48 in the section “Descriptor Area Registers”, and Chapter 5, page 67 in the section “Buffer Descriptor Control (BDCtl)”.

Receive Descriptor Area Initialization

The receive descriptor area is initialized by writing the descriptor area base and limit registers. The controller will use these registers to initiate writing of the receive queue in the receive descriptor area.

Transmitting a Frame

“Transmit Queue Initialization” in Chapter 7 on page 84, describes batch processing transmission. For each batch of frames to transmit, the system initializes the transmit queue, and writes the head of the queue into the transmit frame pointer register.

For continuous polling transmission, the list of frame descriptors is terminated by a dummy frame descriptor, which is owned by the system. When the controller reaches the dummy record, it will enter a polling mode. In this mode, the controller periodically reads the frame descriptor control (FDCtl) field, waiting for the FDOwner bit to be cleared by the system. The frequency of polling is controlled by the transmit polling counter register.

To transmit a frame in continuous polling mode, the system writes a new frame descriptor for the frame to transmit at the tail of the transmit queue. This is done by over-writing the old dummy frame descriptor, creating a new dummy frame descriptor, and setting the next field of the old frame descriptor to the new dummy frame descriptor. The last step of the overwrite is to clear the FDOwner bit of the old frame descriptor, giving ownership to the controller.

Transmit Complete Notification

The system can obtain transmission completion information in a variety of ways:

- Request an interrupt.
- Poll the FDCtl field of transmitted frame descriptors, for system ownership.
- Poll the transmit frame pointer register.

Interrupts can be requested at the end of each frame transmitted, or at the end of selected frames. When polling the transmit frame pointer register, the system call look for an invalid value (batch processing mode), or look for the address of the dummy frame descriptor (continuous polling mode).

Receiving a Frame

To enable the MAC to receive frames, system software must do the following:

1. Initialize the Free Buffer List and Free Descriptor Areas, as described earlier in this chapter on page 84, in the section , “Queue Initialization”.

2. Write a dummy frame descriptor into the free descriptor area, setting the FDOwner bit of the FDCtl field so the controller owns it.

3. Initialize the receive frame pointer register to the address of the dummy frame descriptor in the free descriptor area.

There are two ways that system software can be notified about received frames:

- Request an interrupt for each frame received.
- Poll the dummy frame descriptor, looking for the FDOwner bit to be set.

Interrupts are enabled by setting the completion interrupt enable bit of the receive control register.

Once a frame is received, the system must do the following:

- Process the frame descriptor and free it for reuse at a future time.
- Free buffers as they are returned, and add them to the Free Buffer List.

Processing Received Frame Descriptors

The Free Descriptor Area is intended to be used in a FIFO manner. However, different applications will take different amounts of time to process frames and return associated buffers. So frame descriptors allocated by the controller are copied to another area and freed in the order in which they are received, before being passed up the protocol stack.

Freeing Buffers

There are two ways that buffers may be allocated:

- Starting a new frame in a new buffer.
- Placing several frames or parts of frames in a single buffer.

The allocation mode is controlled by either the buffer fragment size register, or the frame descriptor control field. The single frame mode has the advantage of simpler memory management, but the disadvantage of less efficient memory utilization. The packed buffer mode has the advantage of more efficient memory utilization, but the disadvantage of more complex memory management.

Packed buffers require some additional managing, because of the possibility of multiple frames or fragments of frames in the same buffer area. The controller counts the number of buffers created in the same buffer area, and provides this count as the RxBDSeqN field of the BDCtl field in the Buffer Descriptor. System software can then count returned fragments, until all fragments are returned.

A buffer ID value, RxBDID, is copied from the buffer descriptors in free buffer queue, to the buffer descriptors in the received frame queue. Up to 256 ID values are available. If more are needed, several techniques are available. One alternative is to provide ID extension bits in the FDSystem field. Another is to calculate high order bits from the buffer pointer values, which point into the buffer.

Processing Interrupts

When an interrupt occurs, it is generally on a shared interrupt line. To see if this PCI device is the source of an interrupt, system software reads the Interrupt Source register. Based on the contents of this register, the system software may need to read additional registers, such as the Transmit or Receive Status registers.

This page left blank intentionally

PCI Clock Timing Parameters

The timing diagrams for PCI bus operations can be found in the PCI Specification. Timing diagrams for the MII operations and 10Mbps Ethernet can be found in the IEEE 802.3 standard document. Timing diagrams for the supported EEPROM devices can be found on the respective data sheets.

This section conforms to PCI Local Bus Specification, Section 4.2.3, “Timing Specification.” Figure 8-1 shows the PCI clock waveform and measurement points for both 5V and 3.3V signaling environments.

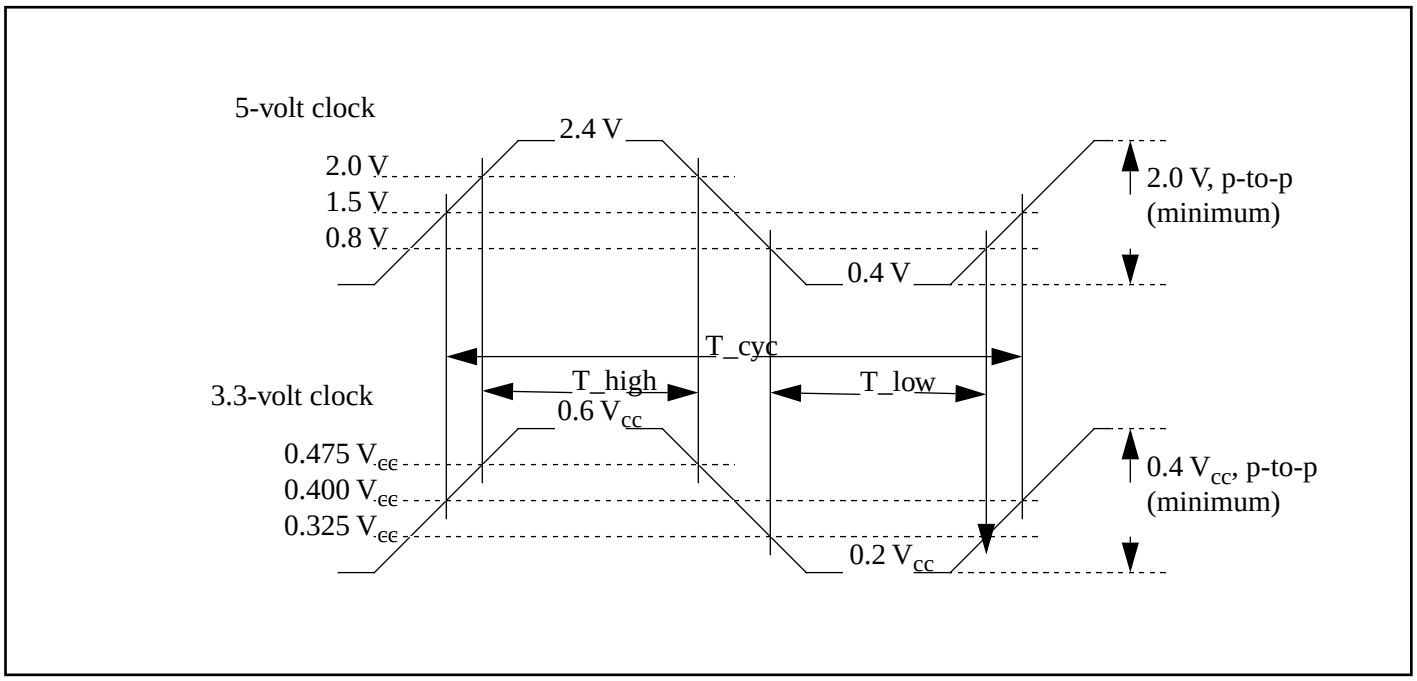


Figure 8-1 PCI Clock Waveforms

In order for the controller to support 100Mbps transmit and receive, the PCI Clk should be operated at between 25 and 33MHz. Clock rates of 16MHz or slower can support 10Mbps operation.

The controller is a fully static design, and can have its clock rate lowered or stopped, providing transmission and reception have been disabled.

Detailed Timing Parameters for each PCI Operation/Transaction

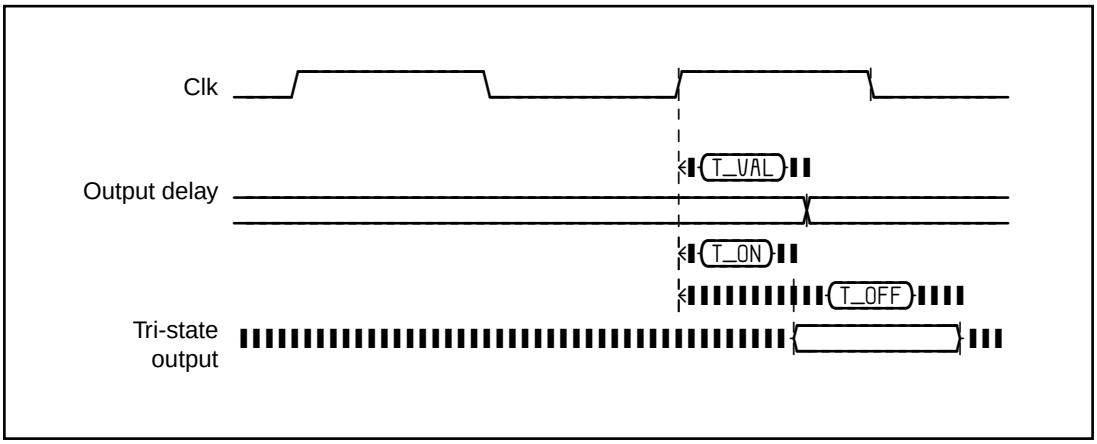


Figure 8-2 Output Timing Measurement Conditions.

High levels are at V_th, low levels are at V_tl, and transitions are at V_test.

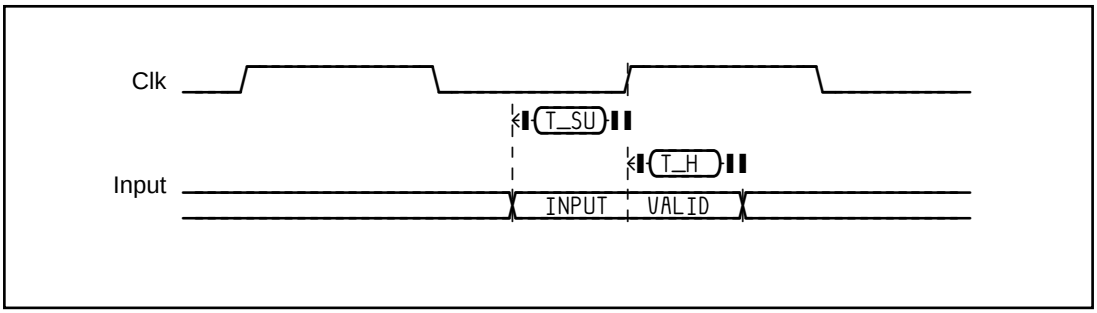


Figure 8-3 Input Timing Measurement Conditions.

High levels are at V_th, low levels are at V_tl, and transitions are at V_test.

Table 8-1 shows the timing parameters for 5V and 3.3V signaling environments.

Table 8-1 5V and 3.3V Timing Parameters

Symbol	Parameter	Min.	Max.	Units	Figure	Notes
t_{val}	Clk-to-signal-valid delay for bussed signals	2	11	ns	8-2	1, 2
$t_{val}(ptp)$	Clk-to-signal-valid delay for point-to-point signals	2	12	ns	8-2	1, 2
t_{on}	Float-to-active delay	2	-	ns	8-2	-
t_{off}	Active-to-float delay	-	28	ns	8-2	-
t_{su}	Input setup time to Clk for bussed signals	7	-	ns	8-3	2
$t_{su}(ptp)$	Input setup time to Clk for point-to-point signals	Gnt#: 10 Req#: 12	-	ns	8-3	2
t_h	Input hold time from Clk	0	-	ns	8-3	-
t_{rst}	Reset active time after power stable	1		ms	-	3
$t_{rst-clk}$	Reset active time after Clk stable	100		µs	-	3
$t_{rst-off}$	Reset active to output float delay	-	40	ns	-	3, 4
t_{rrsu}	Req64# to Rst# setup time	$10*T_{cyc}$	-	ns	-	-
t_{rrh}	Rst# to Req64# hold time	0	50	ns	-	-

Notes:

1. Minimum times are measured with 0 pF equivalent load; maximum times are measured with 50 pF equivalent load. Actual test capacitance may vary, but results should be correlated to these specifications.
2. Req# and Gnt# are point-to-point signals, and have different output valid delay and input setup times than do bussed signals. Gnt# has a setup of 10, and Req#, 12. All other signals are bussed.
3. Rst# is asserted and deasserted asynchronously with respect to Clk.
4. All output drivers must be floated when Rst# is active.

PCI Measurement and Test Conditions

Table 8-2 shows the measurement and test condition parameters for 5V and 3.3V signaling environments.

Table 8-2 Measurement and Test Condition Parameters

Symbol	5V signaling	3.3V signaling	Units	Note
V_{th}	2.4	$0.6\ V_{cc}$	V	1
V_{tl}	0.4	$0.2\ V_{cc}$	V	1
V_{test}	1.5	$0.4\ V_{cc}$	V	-
V_{max}	2.0	$0.4\ V_{cc}$	V	1
Input signal edge rate	1		V/ns	-

Note:

1. The input test for the 5V environment is done with 400 mV of overdrive (over V_{ih} and V_{il}); the test for the 3.3V environment is done with $0.125*V_{cc}$ mV of overdrive. Timing parameters must be met with no more overdrive than this. V_{max} specifies the maximum peak-to-peak waveform allowed for testing input timing.

Detailed Timing Parameters for Each MII Operation/Transaction

This section conforms to Draft Supplement to ANSI/IEEE Std. 802.3, Section 22.3, Signal Characteristics.

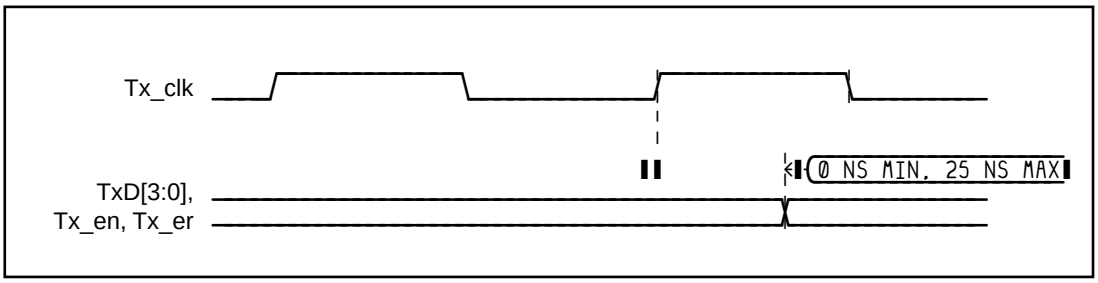


Figure 8-4 Transmit signal timing relationships at the MII

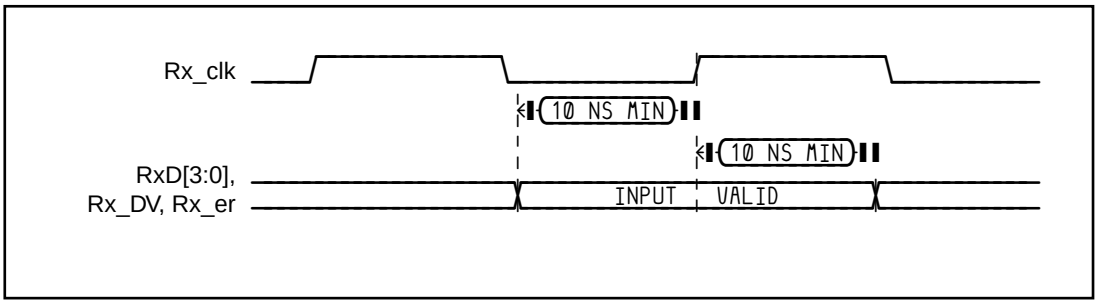


Figure 8-5 Receive signal timing relationships at the MII

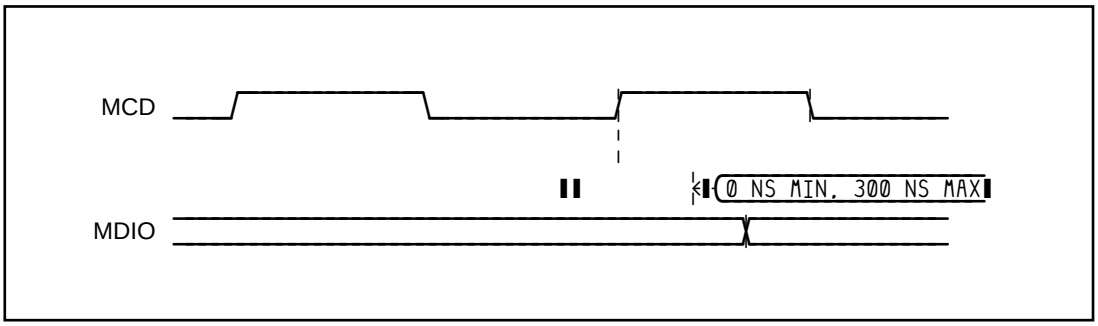


Figure 8-6 MDIO sourced by PHY

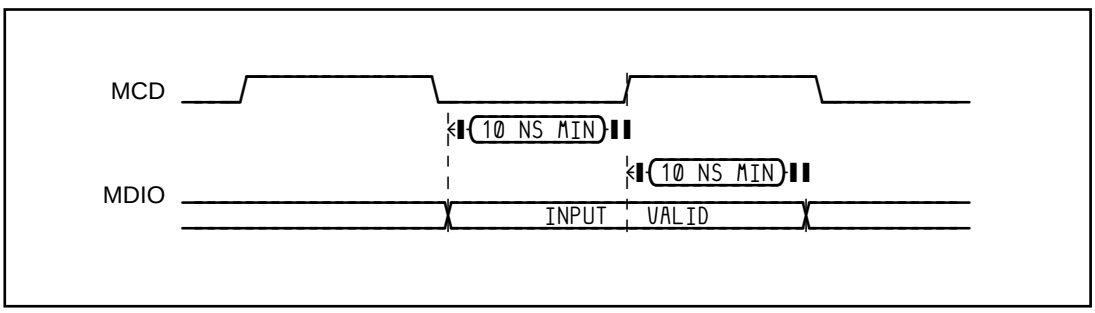


Figure 8-7 MDIO sourced by STA

Table 9-1 Absolute Maximum Ratings

Symbol	Item	Rating
Vdd	Supply Voltage	-0.5V to 4.6V
Vin	Input Voltage	-0.5 to 5.5 V
Tsolder	Soldering Temperature(10 s)	240 °C
Tstg	Storage Temperature	-65 °C to 150 °C
Topr	Operating Temperature	0 °C to 70 °C

Table 9-2 DC Characteristics (PCI Pins)

Symbol	Item	Min.	Typ.	Max.	Unit
Vih(PCI)	Input High Voltage(#5)	0.5Vdd	—	5.25	V
Vil(PCI)	Input Low Voltage	—	—	0.4Vdd	V
Voh	Output High Voltage(IOH=-2.0 mA)	0.9Vdd	—	—	V
Vol	Output Low Voltage(IOL=6 mA)	—	—	0.1Vdd	V
Iil	Input Leakage	-10	—	10	μA

Table 9-3 DC Characteristics (Other PCI Pins)

Symbol	Item	Min.	Type.	Max	Unit	Notes
Vdd	Supply Voltage	3	3.3	3.6	V	—
Iddd	Operating Current (3.3 V, 33 MHz)	—	50	80	mA	—
Iih1	Input High Current 1	30	—	160	μA	1
Iih2	Input High Current 2	-10	—	10	μA	2
Iil1	Input Low Current 1	-160	—	-30	μA	3
Iil2	Input Low Current 2	-10	—	10	μA	4
Vih	Input High Voltage	2	—	5.25	V	5
Vil	Input Low Voltage	—	—	0.8	V	—
Voh	Output High Voltage (IOH=2.5 mA)	2.4	—	—	V	—
Vol	Output Low Voltage (IOL=2.5 mA)	—	—	0.4	V	—

Notes
1. With pull down resistor
2. Non pull down resistor
3. With pull-up resistor
4. Non pull-up resistor
5. All input pins are 5V tolerant.

This page left blank intentionally

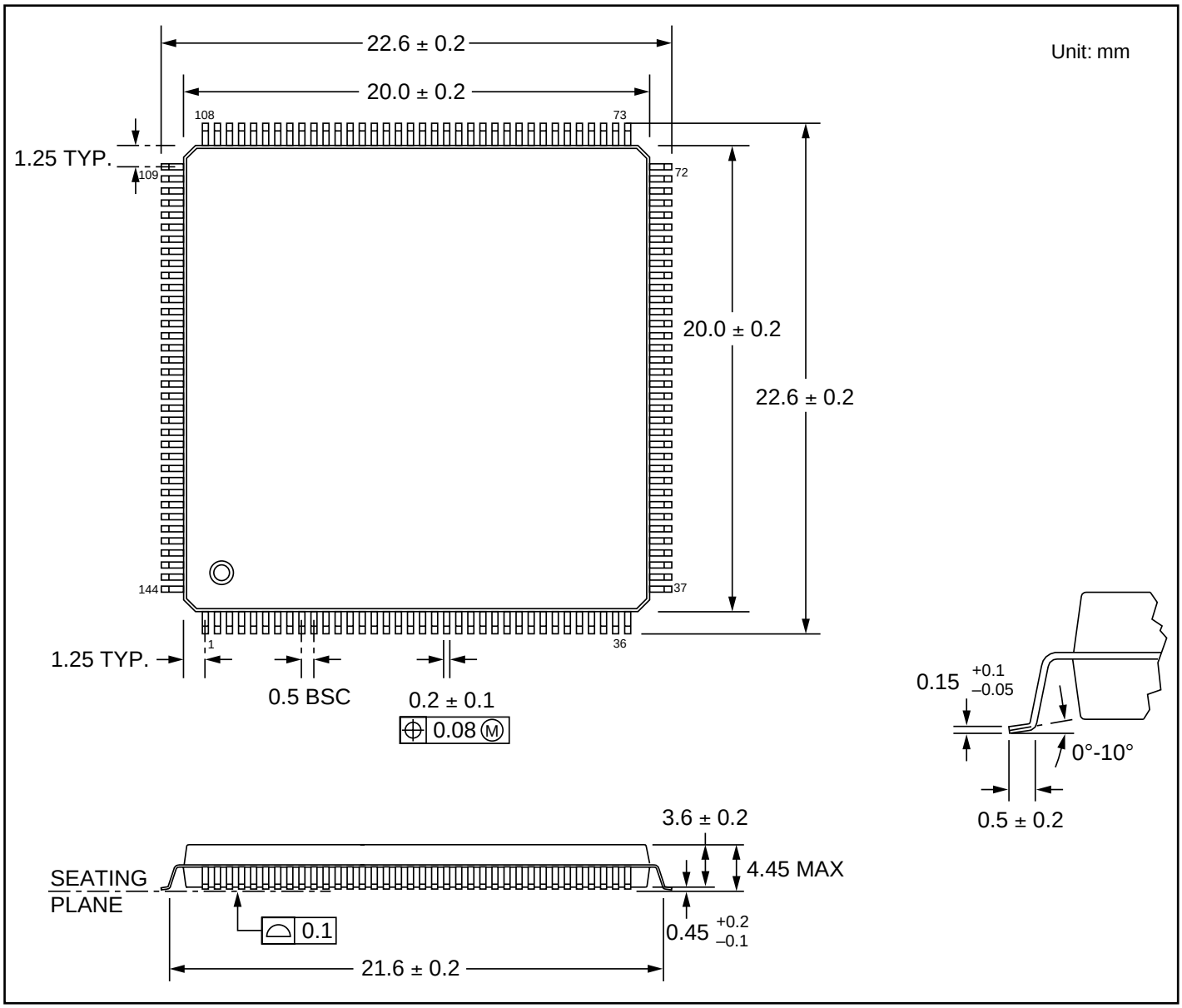


Figure 10-1 QFP144-P-2020-0.50 [144-Pin Quad Flat Package]

Pin Assignments

Table 10-1 Ethernet Controller Pin Assignments

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	VSS	39	AD6	77	PTXD_10
2	AD23	40	VSS	78	VSS
3	AD22	41	AD5	79	PTXC_10
4	AD21	42	AD4	80	VDD
5	AD20	43	AD3	81	VSS
6	VDD	44	VSS	82	VDD
7	AD19	45	AD2	83	CRS
8	AD18	46	AD1	84	COL
9	VSS	47	AD0	85	VSS
10	AD17	48	VDD	86	TXD3
11	AD16	49	VSS	87	TXD2
12	C_BE_L2	50	SCA	88	VSS
13	FRAME_L	51	SCB	89	TXD1
14	IRDY_L	52	VSS	90	VDD
15	TRDY_L	53	PCAM_HIT_L	91	VSS
16	VDD	54	PCAM_LOAD	92	TXD0
17	DEVSEL_L	55	VDD	93	TX_EN
18	VDD	56	NC	94	VDD
19	VSS	57	VSS	95	PTX_CLK
20	STOP_L	58	SCA2	96	VSS
21	PERR_L	59	SCB2	97	TX_ER
22	SERR_L	60	VDD	98	PRX_ER
23	PAR	61	NC	99	PRX_CLK
24	C_BE_L1	62	PPROM_DI	100	PRX_DV
25	VSS	63	PPROM_DO	101	PRXD0
26	AD15	64	VSS	102	VSS
27	AD14	65	PPROM_CLK	103	PRXD1
28	VDD	66	PPROM_CS	104	VDD
29	AD13	67	VSS	105	PRXD2
30	AD12	68	VDD	106	PRXD3
31	VSS	69	PRXC_10	107	PMDC
32	AD11	70	PCRS_10	108	MDIO
33	AD10	71	PRXD_10	109	VDD
34	AD9	72	PLOOP_10	110	VSS
35	AD8	73	VSS	111	SI
36	C_BE_L0	74	PLINK_10	112	SO
37	VDD	75	PCOL_10	113	SI2
38	AD7	76	PTXEN_10	114	SO2

Table 10-1 Ethernet Controller Pin Assignments (continued)

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
115	STM	125	INTA_L	135	AD29
116	VSS	126	VDD	136	VSS
117	VDD	127	VSS	137	AD28
118	VSS	128	PRST_L	138	AD27
119	T1	129	VDD	139	AD26
120	T2	130	PCLK	140	VDD
121	T3	131	GNT_L	141	AD25
122	T4	132	REQ_L	142	AD24
123	VSS	133	AD31	143	C_BE_L3
124	T5	134	AD30	144	IDSEL

Notes:

1. For a description of the active signals, see Chapter 2, "External Signals".
2. NC indicates a no connect.
3. A=Master, e.g., SCA
4. B=Slave, e.g., SCB
5. T1, T3, T4, T5 must be open.
6. T2 must connect to VSS.

This page left blank intentionally

Constants

MAX_DEFERRAL – The maximum number of milliseconds that the MAC will defer before asserting an “excessive deferral” signal.

MAX_CAM_ENTRIES – The maximum number of six-byte addresses that the CAM can hold in direct compare mode.

Table A-1 shows CAM size.

Table A-1 Cam Size

MAX_CAM_ENTRIES	Bytes	Bits
21	126	1008

Buffer Sizes

Toshiba has designed its TC35815CF Flow Control 10/100Mbps Ethernet Controller to use sufficient internal buffering so that the controller can continue to send and receive packets even when the PCI local bus in not available for reasonably long periods of time. Table A-2 shows the relationship between bus latency and buffer sizes.

Table A-2 Bus Latency and Buffer Sizes

Bus Latency	Bytes	Packets
Transmit >140μs	1700	1 - 20
Receive >160μs	1900	1 - 22

The Bus Latency is the maximum amount of time that the PCI local bus might be tied up, or locked by other devices. The PCI Specification uses 10-30μs as a nominal value for bus latency. But in talking with various system designers, Toshiba believes that 40-100μs is a more realistic value for designs. Certainly there is ample evidence that small buffer sizes on some 10Mbps Ethernet controllers leads to the high failure rate of packet transmission and reception on PC-based networks. Transmitters “running dry” is now recognized as a major source of “runt packets” on these networks.

The numbers for Bytes and Bits are the amount of transmit or receive data that can be sent in large packets at 100Mbps.

The packet numbers are calculated using the minimum packet size (64 bytes), but not taking into account the effects of inter-packet gaps or short packet mode. Internal notification FIFOs must be big enough to buffer this number of minimum-size packets for transmission and reception to continue without blocking.

This page left blank intentionally

Glossary

This glossary contains a brief explanation of technical terms, abbreviations, and acronyms used in this document.

Ethernet and Networking Acronyms and Terms

10BASE5 – 500-meter-per-segment Ethernet using half-inch diameter "thick" coaxial cable. Up to 100 nodes.
10BASE2 – 185-meter-per-segment Ethernet, using RG-58 50-ohm "thin" coaxial cable similar to that used for community-antenna television. Up to 30 nodes. Also, Cheapernet or Thinnet.
10BASE-T – 24-gauge, unshielded, twisted-pair 10Mbps Ethernet.
100BASE-FX – 100 Mbps Ethernet using fibre optic cable.
100BASE-T – Twisted-pair 100Mbps Fast Ethernet, either 100BASE-T4 or 100BASE-TX.
100BASE-T2 – 100 Mbps Ethernet using two pairs of category-3, -4, or -5 generic twisted balanced cable.
100BASE-T4 – Intel's proposal for 100Mbps Ethernet using 8B/6T block coding over four pairs of Category-3 unshielded twisted pair. The data runs on three of the four pairs; the fourth pair handles collision detection. The data wires handle a maximum frequency of 25MHz. Also, 4T+ and 4T-Plus.
100BASE-TX – 100Mbps Ethernet using three-level MLT-3 signaling employed by copper FDDI (CDDI) over two pairs of data-grade, Category-5 unshielded twisted pair wiring. Also, 100BASE-X.
100BASE-X – 100Mbps Ethernet using three-level MLT-3 signaling employed by copper FDDI (CDDI) over two pairs of data-grade, Category-5 unshielded twisted pair wiring. Also, 100BASE-TX.
100BASE-VG – 100Mbps Ethernet alternative. See AnyLAN.
AnyLAN – Also 100BASE-VG AnyLAN. Joint H-P and IBM 100Mbps Ethernet alternative and Token Ring proposal. Also, IEEE 802.12.
ADSL – Asymmetric Digital Subscriber Lines. From Bell Communications Research Inc. Uses as few as four-wire twisted pair.
Alignment error –When the Mac receives a frame that is not an integer number of bytes long, and the CRC is invalid. Synonym: framing error. See also, dribble.
ARP – Address Resolution Protocol. Maps an Internet address to a physical address. Not all networks need it.
AUI – Attachment Unit Interface layer of LAN CSMA/CD. Connects the DTE to the MAU. See OSI. Used with thick Ethernet, involving a drop cable.
Babble – Transmit continuously for more than 1500 bytes.[1]

1. Project 802 Local and Metropolitan Area Networks. Draft Guide to ANSI/IEEE Std. 802.3 (CSMA/CS Access Method and Physical Layer Specifications). Systems Topology Technical Advisory Group (SysTAG) Network Guide, Draft 5.2, January 23, 1994. Section 3.5 False Jabber, page 13.

Big Endian – The byte at memory address 0 contains the most-significant bits. Used by IBM 370, Motorola 68000, Pyramid, and TCP/IP protocol header integers.
Bridge – A store-and-forward device connecting physical networks that merely passes all packets. Unlike a gateway, strictly speaking, it performs no protocol conversion, and unlike a router, it does not switch packets to the appropriate network (although some bridges can "learn" where various hosts reside and route packets).
Broadcast address – A distinguished, predefined multicast address that always denotes the set of all stations on a given local area network. All ones in the destination address field shall be predefined to be the broadcast address, ff-ff-ff-ff-ff-ff.
Bundle – A group of signals which have a common set of characteristics and differ only in their information content.
Capture effect – Under heavy load, if at least one of the contending stations is capable of transmitting back-to-back packets continuously, that station can "capture" the network for long periods. This occurs because, once a station has succeeded in transmitting, it zeroes its *attempts* counter, so its next *maxBackOff* will be set to 2. Other stations, however, will continue to increment their *attempts* counters, causing their *maxBackOff* variables to grow exponentially. The network still carries traffic at its full rate (except for second-order effects), but its arbitration has become unfair.
CCITT – Consultative Committee for International Telephony and Telegraphy. Renamed ITU.
Cheapernet – 185-meter-per-segment Ethernet. Also, 10BASE2.
CSMA/CD – Carrier Sense Multiple Access with Collision Detection. Used by Ethernet.
CDDI – Copper Distributed Data Interface. FDDI without the fiber.
CO – Central Office.
CRC – Cyclic Redundancy Code. Called Cyclic Redundancy Check in IEEE 802.3 document.
DCE – Data Communications Equipment.
DS0 – 64-Kbits asynchronous rate. One voice channel.
DS1 – 1.544Mbps asynchronous rate. 24 DS0 signals. T1 equipment and cable carry DS1 rate and format between COs.
DS2 – 6.312Mbps asynchronous rate. Four DS1 signals.
DS3 – 44.736Mbps asynchronous rate. 28 DS1 signals. T3 equipment and cable carry DS3 rate and format between COs.
DMA – Direct Memory Access.
Dribble –When the MAC receives a frame that is not an integer number of bytes long. Dribble produces an alignment error when the CRC is invalid.
DTE – Data Terminal Equipment. If the AUI is not exposed, this is the PLS and up. See OSI.
EFD – The End of Frame Delimiter <EFD> indicates the end of a transmission. For the MII, deassertion of the Tx_en signal constitutes an end of frame delimiter.
Fast Ethernet group – 100Mbps CSMA/CD Ethernet proposal C.f. 100BASE-VG or AnyLAN. Supported by approximately 30 companies, including Grand Junction Networks Inc., Sun Microsystems Inc., 3Com corp., Cabletron Systems Inc., and Synoptics Communications Inc.
FCS – Frame Check Sequence. Also, CRC.
FDDI – Fiber Distributed Data Interface. Fiber-optic Data Distribution Interface. 100Mbps.
FIFO – First-In, First-Out data buffer. Also, silo or queue.
Fragment – A frame that is less than 64 bytes (minFrameSize) long, exclusive of preamble and SFD. Also, runt packet.

Frame – According to 802.3 Tutorial, a frame is everything in a packet except the preamble and the start frame delimiter: destination and source addresses, length, LLC data, padding, and frame check sequence.
Framing error – See alignment error, dribble.
Gateway – A device connecting networks that performs some protocol conversion.
Globally administered address – An Ethernet address whose second bit transmitted, used to distinguish between locally or globally administered addresses, is set to 0, indicating a globally administered (or U, universal) address. If an address is to be assigned locally, this bit shall be set to 1. Note that for the broadcast address, this bit is also a 1.
Group address – An Ethernet destination address whose first bit transmitted, used to identify it either as an individual or as a group address, is 1, indicating that the address field contains a group address that identifies none, one or more, or all of the stations connected to the local area network. More commonly called a multicast address. The first byte of a multicast address is odd, for example, 01-00-00-00-00-00. There are two kinds of multicast address: (a) Multicast-group address. An address associated by higher-level convention with a group of logically-related stations. (b) Broadcast address. A distinguished, predefined multicast address, ff-ff-ff-ff-ff-ff, that always denotes the set of all stations on a given local area network.
HDLC – High-level Data Link Control. Used in X.25.
Heartbeat – Signal Quality Error (SQE).
IAB – Internet Activities Board.
ICMP – Internet Control Message Protocol. The protocol to handle error and control information between gateways and hosts. TCP/IP networking software, not user processes, generates and processes ICMP messages.
IEEE 1149.1 – See JTAG.
IEEE 802.12 – Joint HP and IBM 100Mbps proposal. Also, 100BASE-VG or AnyLAN.
IEEE 802.3 – Information technology–Local and metropolitan area networks–Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. International Standard ISO/IEC 8802-3: 1993(E), ANSI/IEEE Std. 802.3, 1993 Edition, July 8, 1993. Also, Ethernet.
IEEE 802.4 – Token bus.
IEEE 802.5 – Token ring.
Individual address – An Ethernet address whose first bit transmitted, used to identify it either as an individual or as a group address, is 0, indicating that the address field contains an individual address, and so is associated with a particular station on the network. The first byte of a individual address is even, for example, 00-00-00-00-00-00.
IP – Internet Protocol. Fragments (segments), routes, delivers, and reassembles packets for TCP, UDP, and ICMP. Connectionless and unreliable. Adds a 20-byte header and a check sum.
ISDN – Integrated Services Digital Network.
ISO – International Organization for Standardization.
Jabber – In order to protect the network from babbling nodes, IEEE Std. 803.3 requires MAUs to inhibit transmission onto the network if they have been transmitting for an excessive time. The window for jabber must be between 20 and 150ms.[1]

1. Project 802 Local and Metropolitan Area Networks. Draft Guide to ANSI/IEEE Std. 802.3 (CSMA/CS Access Method and Physical Layer Specifications). SysTAG Network Guide, Draft 5.2, January 23, 1994. Section 3.5 False Jabber, page 13.

JTAG – Joint Test Action Group. A group of companies that developed what became IEEE 1149.1 Test Access Port... for board-level production test of integrated-circuit pin continuity.
LAN – Local Area Network.
Little Endian – The byte at memory address 0 contains the least-significant bits. Used by Intel x86, DEC Vax, and DEC PDP-11.
LLC – Logical Link Control layer of LAN CSMA/CD. The upper half of the OSI (which see) reference model data link layer, between the MAC and the network layer.
Locally administered address – An Ethernet address whose second bit transmitted, used to distinguish between locally or globally administered addresses, is set to 1, indicating a locally administered address. If an address is to be assigned globally (or U, universal), this bit shall be set to 0. Note that for the broadcast address, this bit is also a 1.[1]
MAC – Media Access Control layer of LAN CSMA/CD. The lower half of the OSI (which see) reference model data link layer, between the LLC and the PLS.
MAU – Medium Attachment Unit layer of LAN CSMA/CD. See OSI.
MDI – Medium Dependent Interface layer of LAN CSMA/CD. The part of the MAU that connects the PMA to the medium. See OSI.
MII – Media Independent Interface. A four-bit-wide interface between a reconciliation layer and the PLS. The reconciliation layer connects to the MAC using the existing 802.3 MAC-PLS interface.
Minimum frame length – 64 bytes (512 bits). Note that this does not include the preamble and the start frame delimiter.
Multicast address – An Ethernet destination address whose first bit transmitted, used to identify it either as an individual or as a group address, is 1, indicating that the address field contains a group address that identifies none, one or more, or all of the stations connected to the local area network. The first byte of a multicast address is odd, for example, 01-00-00-00-00-00. There are two kinds of multicast address: (a) Multicast-group address. An address associated by higher-level convention with a group of logically-related stations. (b) Broadcast address. A distinguished, predefined multicast address that always denotes the set of all stations on a given local area network.
Multicast-group address – An address associated by higher-level convention with a group of logically-related stations. This is one of two kinds of multicast address, the other being the broadcast address, ff-ff-ff-ff-ff-ff.
NDIS – Network Driver Interface Specification for a generic device driver, independent of protocol or hardware.
Network acquisition time – If the DTE transmits for a period exceeding the net acquisition time without detecting a collision, then the DTE is said to have acquired the network. The DTE will send the remainder of its frame without the possibility of having a collision in a correctly operating network. By this time, all DTEs in the network have detected network activity and are deferring to it.... For example, the network acquisition time on a single-segment 10BASE5 network of 500-meter length is approximately 108 bit times.[2] Also called round trip delay.
NIC – Network Interface Controller or Card. Also Network Information Center, at SRI International, which assigns a network a class field (A, B, or C) and a unique network ID field of the 32-bit TCP/IP addresses.

1. ANSI/IEEE Std. 802.3, 1993 Edition. Section 3.2.3 Address Fields, page 42.
2. Project 802 Local and Metropolitan Area Networks. Draft Guide to ANSI/IEEE Std. 802.3 (CSMA/CS Access Method and Physical Layer Specifications). SysTAG Network Guide, Draft 5.2, January 23, 1994. Section 2.3.1: Network Acquisition Time, page 4.

OC1 – 51.84Mbps Optical Carrier 1, SONET Synchronous Transport Signal STS-1. 28 DS1 signals or one DS3.
OC3 – 155.52Mbps Optical Carrier 3, SONET Synchronous Transport Signal STS-3. Three byte-interleaved STS-1 signals.
OC12 – 622.08Mbps Optical Carrier 12, SONET Synchronous Transport Signal STS-12. Twelve byte-interleaved STS-1 signals.
OC48 – 2488.32Mbps Optical Carrier 48, SONET Synchronous Transport Signal STS-48. 48 byte-interleaved STS-1 signals.
Octet – byte.
OSI – Open System Interconnection reference model layers adopted by the **ISO** – application, presentation, session, transport, network, data link (composed of LLC and MAC layers of LAN CSMA/CD), and physical (composed of PLS and PMA layers of LAN CSMA/CD, connected by AUI).
Packet – According to 802.3 Tutorial, a packet is a frame, preceded by the preamble and the start frame delimiter fields. Thus, it consists of (1) preamble, (2) start frame delimiter, (3) destination and (4) source addresses, (5) length, (6) LLC data, (7) padding, and (8) frame check sequence fields.
PCI – Peripheral Component Interconnect. 32 or 64 bits wide, 0 to 33MHz clock rate, 132 to 264 MByte/s (peak). Intel Corporation released version 1.0 of the PCI specification on June 22, 1992, and the PCI Special Interest Group released revision 2.0 on April 30, 1993.
PING – Packet InterNet Groper. Tests the reachability of another site by sending an ICMP echo request message.
PCMCIA – Personal Computer Memory Card International Association.
PDU – Protocol Data Unit in ATM.
PHY - Physical Layer Entity, as defined in the 802 Architecture and Overview Standard. The word PHY is used to denote the set of functions associated with a physical layer protocol.
PLS – Physical Signaling layer of LAN CSMA/CD. See OSI.
PMA – Physical Medium Attachment. See OSI.
PMD – Physical Medium Dependent layer. PMD connects to MAC via MII on one side, and connects to MDI on the other.
PPP – Point-to-Point Protocol. Successor to SLIP, Serial Line Internet Protocol. Provides router-to-router and host-to-network connections over both synchronous and asynchronous circuits.
PSTN – Public Switched Telephone Network.
Preamble – The first field of a packet, up to seven bytes long. Each byte has value 10101010, transmitted left to right.
QAM – Quadrature Amplitude Modulation. Modulation of both amplitude and phase, to increase the information capacity of a channel.
RARP – Reverse Address Resolution Protocol. Maps a physical address to an Internet address. Only some networks need it.
Repeater – Connects two or more Ethernet segments, with signal amplification and timing and preamble regeneration, but without storing packets.
RJ-11 – Four-wire modular telephone connector.
RJ-45 – Eight-wire modular telephone connector.
Router – A store-and-forward protocol-dependent device connecting networks that switches packets to the appropriate network.

Runt – A frame that is less than 64 bytes (minFrameSize) long, exclusive of preamble and SFD. Synonym: fragment.
Saturn – Sonet-ATM User Network. A group to develop chips to Sonet and ATM Forum specifications.
SCSI – Small Computer System Interface.
SDH – Synchronous Digital Hierarchy. ITU's European designation for fiber-optic transmission. See also Sonet.
SDLC – Synchronous Data Link Control. Used in Systems Network Architecture.
SFD – Start Frame Delimiter. A single byte with value 10101011, transmitted from left to right. It follows the preamble.
Signal Quality Error – SQE. Also, heartbeat.
SLIP – Serial Line Internet Protocol. Predecessor to PPP.
SMDS – Switched Multimegabit Data Service, a 45Mbps cell-relay service based on IEEE 802.6 Distributed Queue Dual Bus.
SMTP – Simple Mail Transfer Protocol. An application program provided by almost every TCP/IP implementation.
SNMP – Simple Network Management Protocol. Allows a TCP/IP host to query other nodes for network-related statistics and error conditions.
Sonet – Synchronous Optical Network. ANSI U.S. fiber-optic transmission standard. See also SDH.
Sonic™ – National Semiconductor Corporation's DP83932B Systems-Oriented Network Interface Controller.
Spanning tree – An algorithm to create a logical topology connecting all network segments, and ensures that only one path exists between any two stations.
SPOOL – Simultaneous Peripheral Operation On-line.
SQE – Signal Quality Error. Also, heartbeat.
Start Frame Delimiter – see SFD.
Station address – An Ethernet address whose first bit transmitted, used to identify it either as an individual or as a group address, is 0, indicating that the address field contains an individual address, and so is associated with a particular station on the network. The first byte of a multicast address is even, for example, 00-00-00-00-00-00.
STS-1 – Synchronous Transport Signals.
T1 – 1.544Mbps.
TAP – Test Access Port. See JTAG.
TCP –Transmission Control Protocol, a connection-oriented, reliable, full-duplex, virtual circuit byte-stream facility for a user process. Uses IP. Part of the TCP/IP protocol suite.
TCP/IP –Transmission Control Protocol/Internet Protocol, the DARPA Internet protocol suite.
Type 1 cable – Shielded, two-pair cable.
Type 3 cable – Unshielded, twisted-pair cable.
UDP – User Datagram Protocol, a connectionless, unreliable datagram facility for a user process. Uses IP. Part of the TCP/IP protocol suite.
Universal address – An Ethernet address whose second bit transmitted, used to distinguish between locally or globally administered addresses, is set to 0, indicating a globally administered (or U, universal) address. If an address is to be assigned locally, this bit shall be set to 1. Note that for the broadcast address, this bit is also a 1.[1]

1. ANSI/IEEE Std. 802.3, 1993 Edition. Section 3.2.3 Address Fields, page 42.

V.32terbo modem – 19.2 Kbits/s.
V.34 modem – 28.8 Kbits/s.
VESA – Video Electronics Standards Association, an association of companies involved in the design and manufacture of video graphics adapters.
VL – VESA Local bus.

This page left blank intentionally

The following features are not supported by the Toshiba TC35815CF Flow Control 10/100Mbps Ethernet Controller.

Peripheral Component Interconnect (PCI) Functions

Some optional PCI functions are not supported.

The controller does not support the Lock# signal, which supports atomic operations. Nor does it support the Interrupt B, C, and D signals, IntB#, IntC#, and IntD#, for multi-function devices, or snoop backoff, SBO#, and Snoop done, SDone, for cache support.

The controller does not support the optional pins for a 64-bit AD data path.

The controller does not support 64-bit addressing for system memory space. Thus, in Chapter 4, page 40, the section named “I/O and Memory Base Address Registers,” only 00 and 01 are supported as values for the Loc field.

IEEE 802.3 Features

The controller does not support the 16-bit option for destination and source addresses.

This page left blank intentionally

TAEC Regional Sales Offices

NORTHWEST
San Jose, CA
TEL: (408) 526-2400
FAX: (408) 526-2410

Portland, OR
TEL: (503) 629-0818
FAX: (503) 629-0827

SOUTHWEST
Irvine, CA
TEL: (949) 453-0224
FAX: (949) 453-0125

Dallas, TX
TEL: (972) 480-0470
FAX: (972) 235-4114

CENTRAL
Chicago, IL
TEL: (847) 945-1500
FAX: (847) 945-1044

NORTHEAST
Edison, NJ
TEL: (732) 248-8070
FAX: (732) 248-8030

Boston, MA
TEL: (781) 224-0074
FAX: (781) 224-1096

SOUTHEAST
Atlanta, GA
TEL: (770) 931-3363
FAX: (770) 931-7602

www.toshiba.com/taec